TechV_28335

使用说明书

北京达盛科技有限公司

第二版：**2009-5-31, Edited By Wjp**

地址：北京市海淀区长春桥路 5 号新起点大厦 2-1501
邮编：100089
电话：010-82564899
Web：http://www.techshine.com/

目　录

第一章　硬件资料介绍

1.　Techv_28335CPU 板介绍

一、 F28335EVM 概况

F28335EVM 是一个独立的嵌入式应用板卡，用户可以通过它直接验证自己的算法，或在此基础上进行最终产品的集成或开发。板卡上面丰富的资源能够满足大多数应用场合的需求，高容量的存储器能够满足各种应用代码的调试。完全的信号扩展使用户更方便进行二次开发。该产品灵活方便的外部接口，可以作为工业控制特别是电机控制系统的集成的配套产品。

选用该板卡可以大大降低系统的研发周期和风险，由于其高的可靠性，为设备生产厂家和最终用户提供了可靠的平台。同时为了简化代码开发，缩短编辑调试时间，提供了方便的接口和大容量的片上 RAM 存储器。并可以使用 C 编辑器进行代码的调试。

1. F28335EVM 主要特点：

- 高性能 32 位浮点 TMS320F28335 数字信号处理器，系统周期为 6.67ns 的处理速度，运算速度可达到 150MIPS。
- 16 路 12bitA/D 转换器，内建 2 个采样保持器，最快的转换速度（S/H+转换）为 80ns；
- 内部存储器：68K SRAM，256K*16bit 以及内建 128K 可加密的 FLASH，编程电压为 3.3V
- 板上零等待 64K*16bit SRAM
- 双路 RS-232 增强型主机通信接口可实现异步通信
- CAN 总线接口，可做终端点或其他任意节点
- 2 个增强型多通道缓寸串口（McBSP）；
- 板上 30MHZ 集成晶振；
- 可编程 4 位拨位开关，4 位可独立编程的指示灯；
- 4 个数据、地址、I/O 以及控制信号扩展接口，扩展所有 DSP 的功能引脚；
- 18 路 PWM 输出
- 提供 USB2.0 接口，方便与计算机传输数据
- 板上 IEEE1149.1JTAG 连接器接口
- 9pin 串口电缆（板上接口是（“male”)；
- 可单电源 5V 供电，也可通过扩展接口从扩展板

2.　Techv_28335CPU 板资源介绍

主要介绍 F28335 评估板的主要部件及其使用方法。并提供相应的接口信息。

F28335EVM 主要由以下几个部分组成：

- Ø F28335 扩展存储单元
- Ø LED 和拨码开关
- Ø 复位和电源电路
- Ø 片上 CAN 总线接口单元
- Ø 片上 RS232 串行接口单元
- Ø JTAG 仿真器接口单元
- Ø CPLD 逻辑译码单元
- Ø USB 通信接口单元
- Ø 扩展接口（J2，J4，J6，J12）信息

1. **F28335 扩展存储单元**

EVM 包含零等待周期的一片 64k*16bit 数据存储器，芯片型号为 ISLV6416。总计提供 64k 的外部存储器。在系统开发过程中，内部的存储器往往要优先使用，这样可以更好地提高系统的运行效率。

此外，外部存储器的速度要受等待周期的影响。F28335 能够内部产生外部接口(XINTF)的等待周期。片外等待周期由片内等待状态产生寄存器确定。为了能够获得零等待的存储器接口，系统必须正确的配置等待寄存器。也可以通过外部 ready 信号来产生等待周期。

外部存储器接口译码通过 XC95144XL 来完成。

F28335 的存储器采用统一编址方式，存储器映射空间如下图：

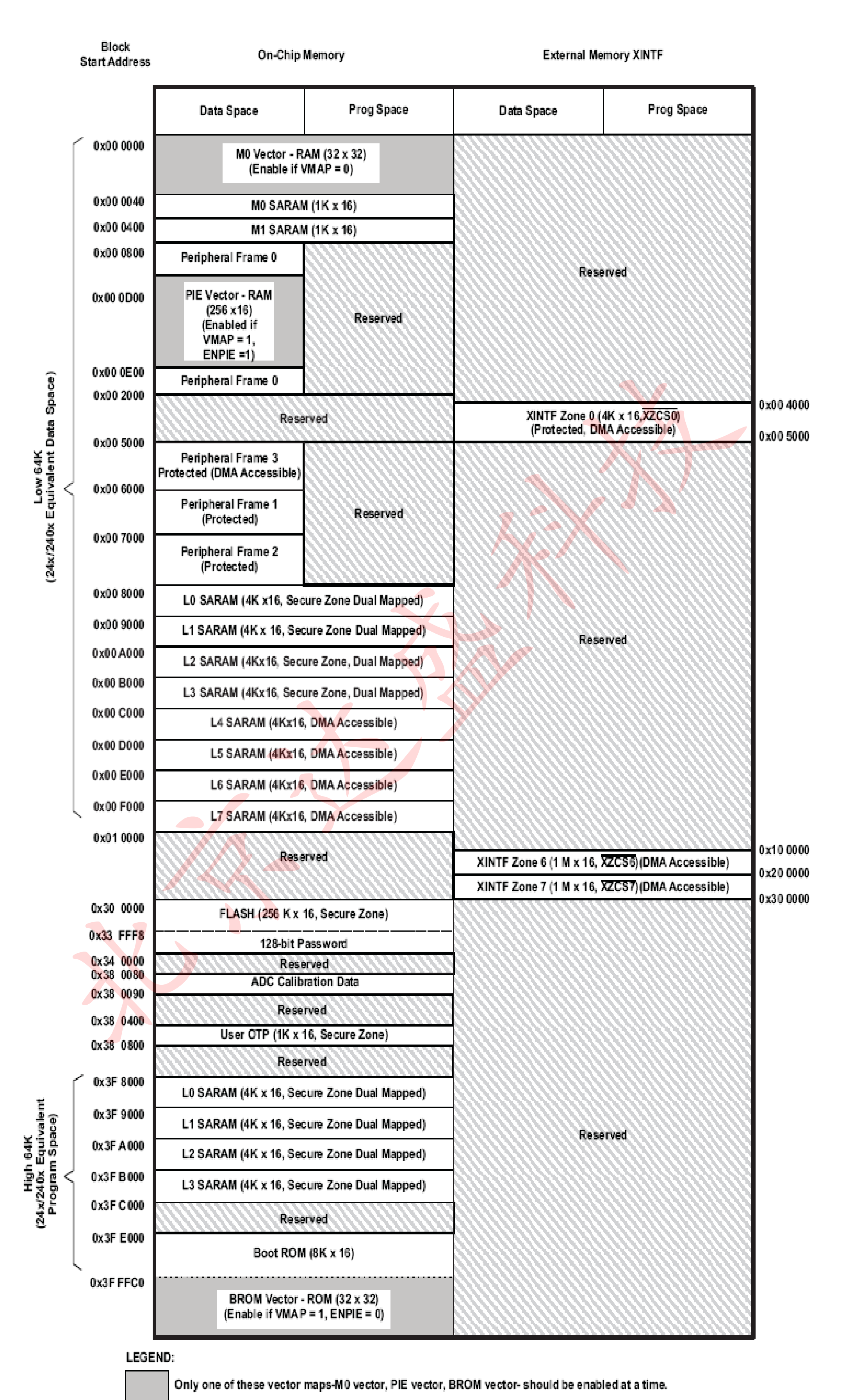


Figure 3-2. F28335/F28235 Memory Map

评估板存储空间分配：

板上扩展存储器预分配如下：

SRAM 存储器：0x100000---0x10ffff，共 64K （U2）

片内存储器地址范围（数据或程序空间）：

M0: 0x0000~0x0400;(1K*16bit)

M1: 0x0400~0x0800; (1K*16bit)

低 64K*16bit SARAM:

L0: 0x008000~0x009000; (4K*16bit)

L1: 0x009000~0x00a000; (4K*16bit)

L2: 0x00a000~0x00b000; (4K*16bit)

L3: 0x00b000~0x00c000; (4K*16bit)

L4: 0x00c000~0x00d000; (4K*16bit)

L5: 0x00d000~0x00e000; (4K*16bit)

L6: 0x00e000~0x00f000; (4K*16bit)

L7: 0x00f000~0x010000; (4K*16bit)

高 64K*16bit SARAM:

L0： 0x3f8000~0x3f9000; (4K*16bit)

L1: 0x3f9000~0x3fa000; (4K*16bit)

L2: 0x3fa000~0x3fb000; (4K*16bit)

L3: 0x3fb000~0x3fc000; (4K*16bit)

OTP: 0x380400~0x380800; (1K*16bit)

FLASH: 0x300000~0x3Ffff8; (256K*16bit)

2. 用户开关和 LED 说明

TMS320F28335 EVM 有 2 个开关，4 个 LED 测试灯和 2 个 LED 指示灯，这些设备都接到 CPLD 上，用户可以通过编程来设定其功能。

拨码开关 SW1 设置：（用户可编程的可控开关）

备注：下表中： “1”表示 ON；“0”表示 OFF

SW2.4	SW2.3	SW2.2	SW2.1	说明
1	1	1	1	MFSRA_D20,MCLKRA_D21,MFSXA,ADCSOCAO,ADCSOCBO 管脚状态不受 CPLD 影响
1	1	1	0	输出外部中断 2 到 MFSRA_d20 管脚
1	1	0	1	输出外部中断 3 到 MFSRA_d20 管脚
1	1	0	0	输出外部中断 NMI 到 MCLKRA_d21 管脚
1	0	1	1	用户可根据自己的应用自行编写程序进行 SW1 开关

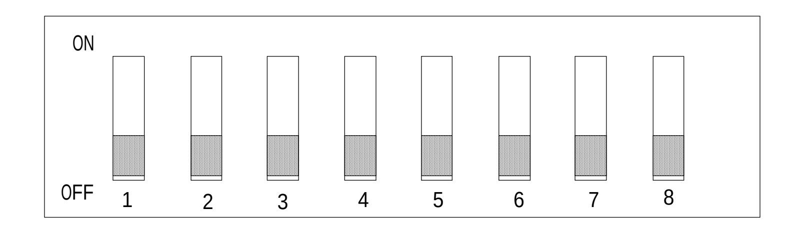
<table>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td rowspan="11">的选择

MFSRA_D20,MCLKRA_D21,MFSXA,ADCSOCA0,ADCSOCB0
管脚状态不受 CPLD 影响</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="5">建议：　　将 SW1 开关全部拨为:SW1.1:OFF; SW1.2:OFF; SW1.3:OFF; SW1.4:OFF</td></tr>
</table>

拨码开关 SW2 的设置：（BOOT 模式选择开关）

SW2.4	SW2.3	SW2.2	SW2.1	BOOT MODEL
1	1	1	1	Jump to Flash <- "boot to Flash"
1	1	1	0	SCI-A boot
1	1	0	1	SPI-A boot
1	1	0	0	I2C-A boot
1	0	1	1	eCAN-A boot
1	0	1	0	McBSP-A boot
1	0	0	1	Jump to XINTF x16
1	0	0	0	Jump to XINTF x32
0	1	1	1	Jump to OTP
0	1	1	0	Parallel GPIO I/O boot
0	1	0	1	Parallel XINTF boot
0	1	0	0	Jump to SARAM
0	0	1	1	Branch to check boot mode
0	0	1	0	Boot to flash, bypass ADC cal
0	0	0	1	Boot to SARAM, bypass ADC cal
0	0	0	0	Boot to SCI-A, bypass ADC cal

拨码开关如下：



LED 说明

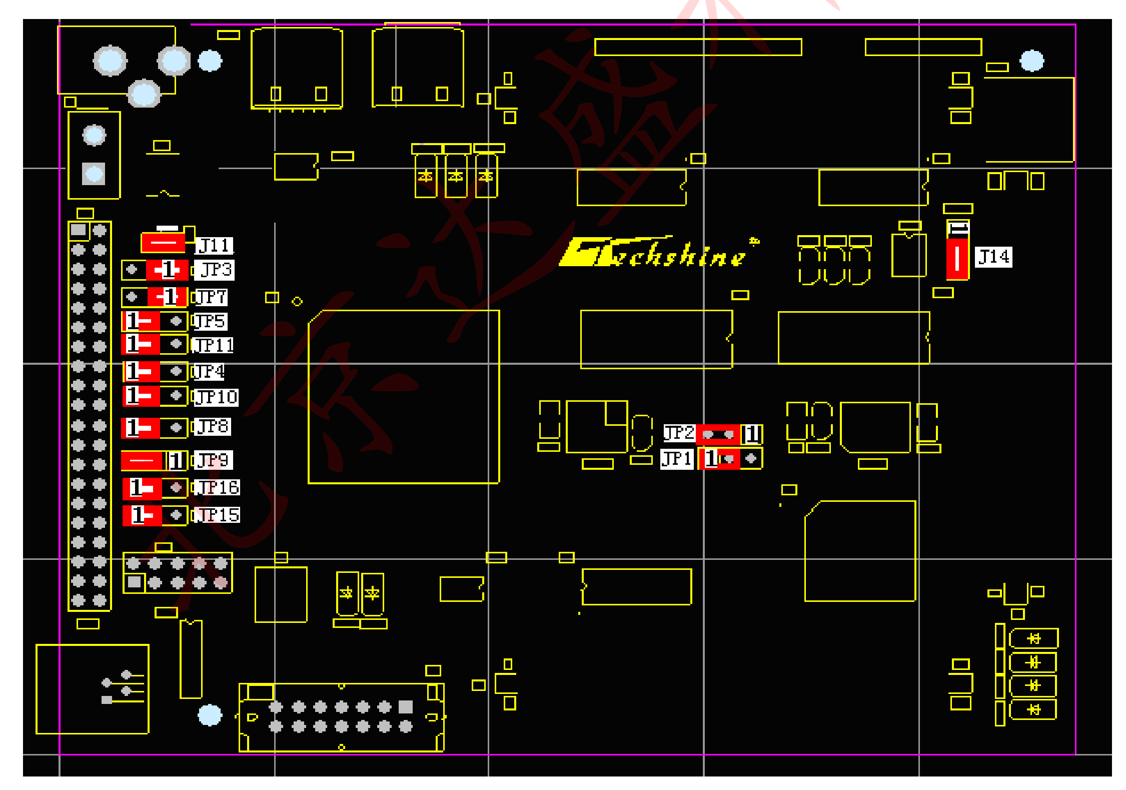
- LED8：可控编程测试灯
- LED7 ：可控编程测试灯
- LED6：可控编程测试灯
- LED5：可控编程测试灯
- LED4：复位信号指示灯
- LED3：DSP 的工作时钟，有时钟信号则闪烁，并且闪 烁频率随 DSP 工作时钟的改变而改变；
- LED2：外围 3.3V 电源指示，亮表示外围 3.3V 电源供电正常；
- LED1：DSP 的 1.9V 电压指示，亮表示 1.8V 供电正常（亮度较暗）；
- LED0：是 DSP 的 5.0V 电压指示，亮表示 3.3V 供电正常。
- 注：以上 LED(除了 LED0、LED1、LED2)的功能都是通过 CPLD 编程实现的，用户可以编程改变这些灯的功能。

3. **功能跳线说明**

TechV_28335 CPU 板上功能跳线说明			
JP1	JP1-1	JP1-2	JP1-3
	NC	ADCREFIN	ADCREFIN_EXT
JP2	JP2-1	JP2-2	JP2-3
	AGND	ADCLO	ADCLO_EXT
JP3	JP3-1	JP3-2	JP3-3
	GND	接 L1	NC
JP4	JP4-1	JP4-2	JP4-3
	EPWM6A	接 28335 第 19 管脚	ADCSOCBO
JP5	JP5-1	JP5-2	JP5-3
	SCLA	接 28335 第 75	EPWMSYNCO
JP7	JP7-1	JP7-2	JP7-3
	接 28335 第 73 管脚	接 U4 第 33 管脚	接 28335 第 90 脚
JP8	JP8-1	JP8-2	JP8-3
	接 28335 第 91 管脚	接 U4 第 31 管脚	接 28335 第 88 管脚
JP9	JP9-1	JP9-2	JP9-3

JP10	JP10-1	JP10-2	JP10-3
	EPWM5A	接 28335 第 19 管脚	ADCSOCBO
JP11	JP11-1	JP11-2	JP11-3
	SDAA	接 28335 第 74 管脚	EPWMSYNCI
JP14	JP14-1	JP14-2	JP14-3
	接 U12 的第一管脚	GND	NC
JP15	JP15-1	JP15-2	JP15-3
	CANRXB	接 28335 第 28 管脚	TZ6
JP16	JP16-1	JP16-2	JP16-3
	CANTXB	接 28335 第 27 脚	TZ5
备注:	详细的管脚定义说明请参看 TechV_28335 原理图和 DATESHEET 说明		

功能跳线在 CPU 板上的位置示意图分布:



出厂前跳动线设置如上:
红色的部分为跳线帽的位置

备注: 具体的跳线应用请以跳线说明为依据
具体的跳线位置请以参看原理图为依据

4． **复位和电源电路单元**

备注：由于 TMS320F28335 功耗比 2812 较大，在片内运行 FLASH 程序时，电源芯片 LM117 和 NCP5663DSA 芯片略微有点发烫属于正常现象。

复位电路包括上电复位和硬件手动复位，每次复位要求至少要有8到10个系统时钟，还要考虑上电晶振的稳定时间。下图为系统的复位电路图。

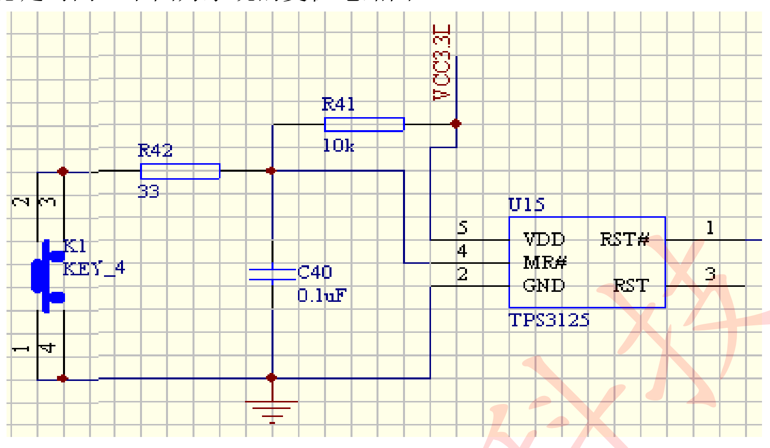
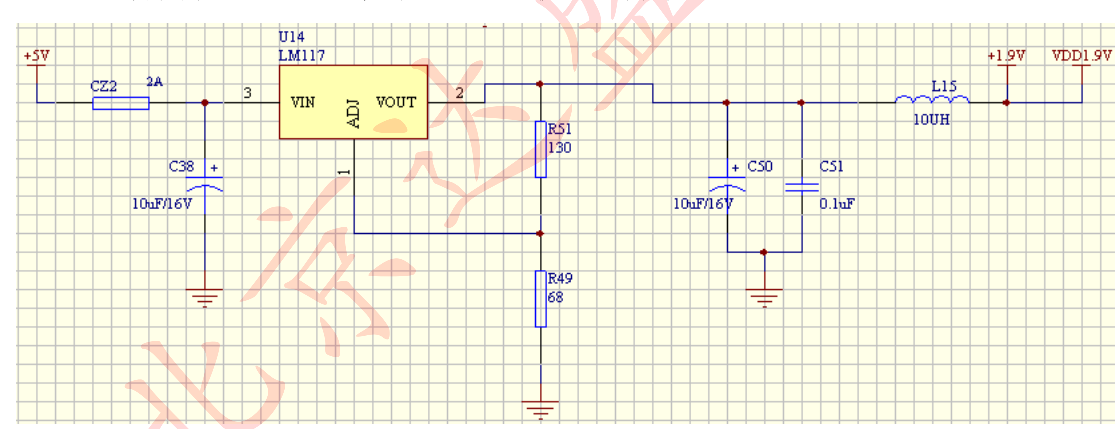
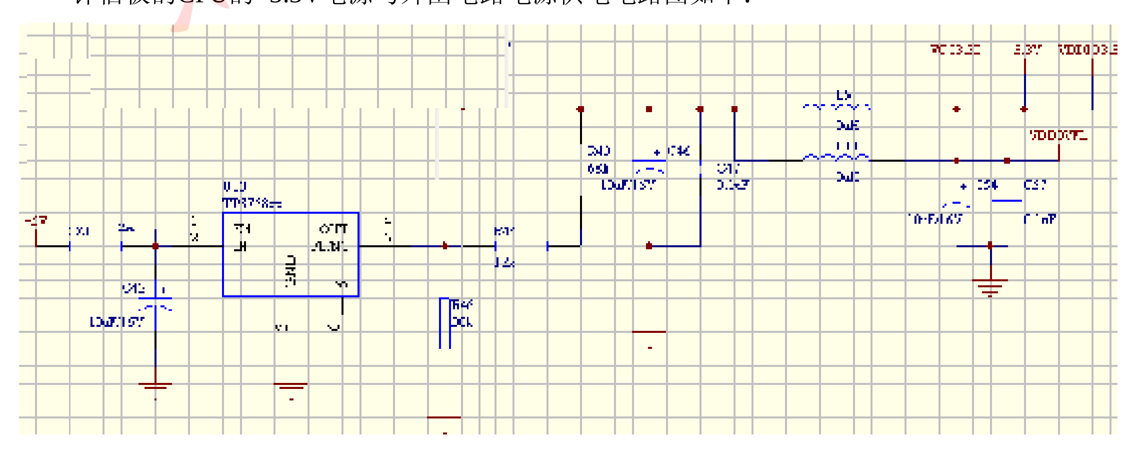


图4、复位电路

F28335采用3.3 V和1.9V双电源供电。本系统采用数字模拟地分离设计。电压转换电路将输入的5V电压转换为3.3V和1.9V。其中+1.9V电压供电电路图如下。

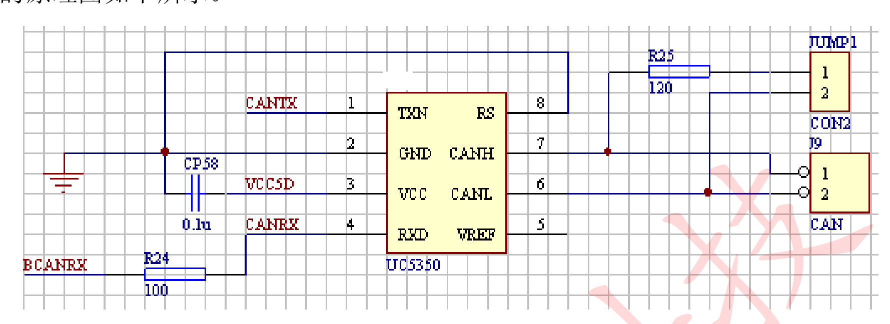


评估板的CPU的+3.3V电源与外围电路电源供电电路图如下：



5. 片上 CAN 总线接口单元

EVM320F28335 有一个 CAN 总线接口，用来提供高速串行通信，本评估板的 CAN 总线接口经过 CAN 总线接口芯片 UC5350 与外部接口相连。板上设计了终端节点信号反射回收电阻 R25，默认情况通过短接跳线 JUMP1 来接入此电阻，因此默认为该模块为终端节点。如果不选用终端节点，则应将 JUMP1 的跳线帽去掉，其中 R25 为 120 欧姆。J9 为 CAN 的一个接收终端。此部分的原理图如下所示。



6. 片上 RS232 串行接口单元

TMS320F28335DSP在片上有两个异步串行通信接口。通过板上的J10同外部主机或其他设备进行通信。

接口采用电话线（male）接头，其中1、4脚接地，2脚是TXD脚，3脚是RXD脚。

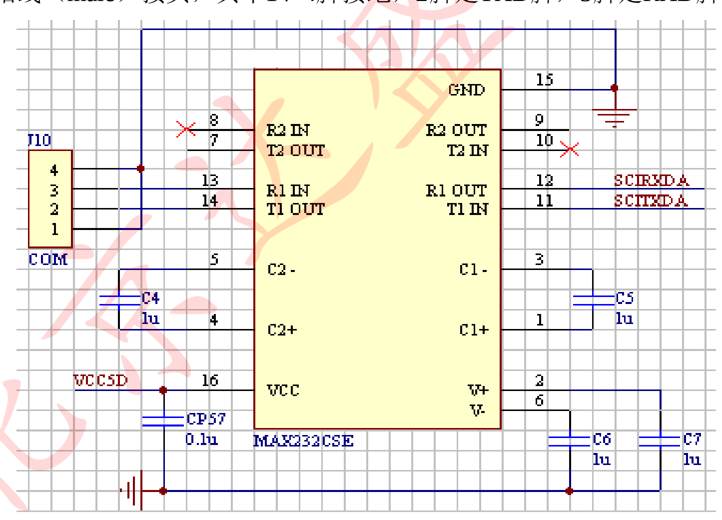


图 6、串行通信接口信号定义

7. JTAG 仿真器接口单元

TMS320F28335评估模块支持14 pin JTAG的接口，引脚分配如图5所示：

Signal	Pin	Pin	Signal
TMS	1	2	TRST-
TDI	3	4	GND
PD (+5V)	5	6	no pin (key)
TDO	7	8	GND
TCK-RET	9	10	GND
TCK	11	12	GND
EMU0	13	14	EMU1

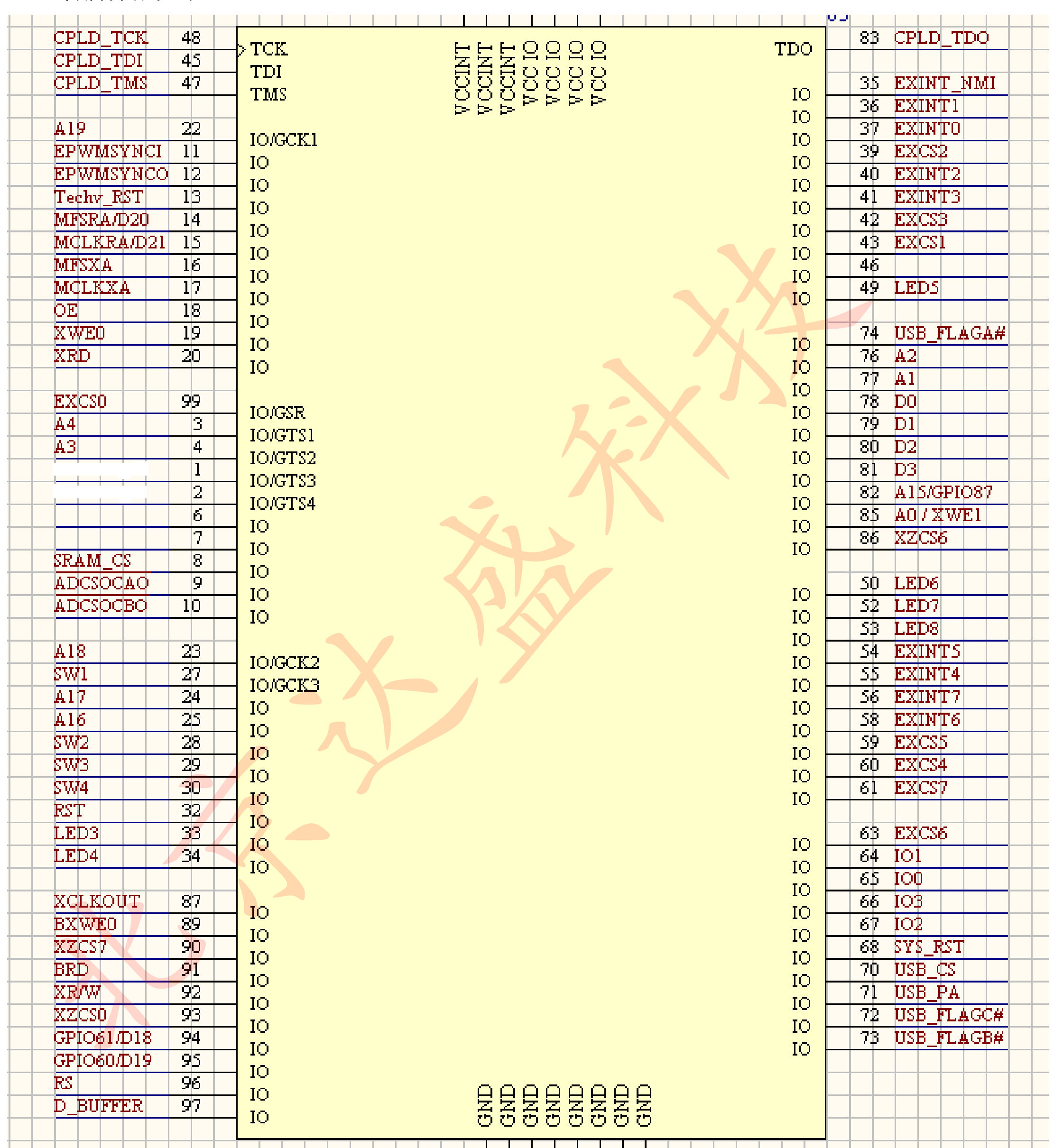
Header Dimensions
Pin-to-Pin spacing, 0.100 in. (X,Y)
Pin width, 0.025-in. square post
Pin length, 0.235-in. nominal

图5、JTAG总线接口

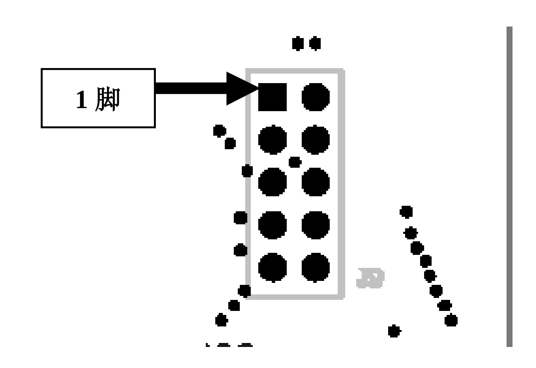
8．CPLD 逻辑译码单元

采用XILINX公司的XC95144XL芯片，完成译码和时序控制。JTAG2为CPLD下载接口。可用XILINX公司的软件，通过并口下载电缆对CPLD在线编程。

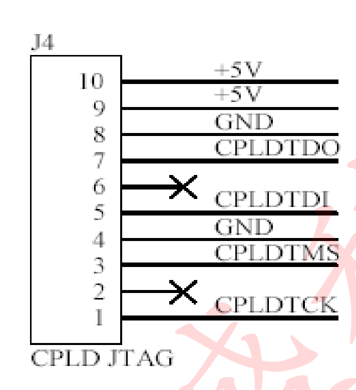
CPLD管脚分配如下：



CPLD JTAG封装如下图所示：



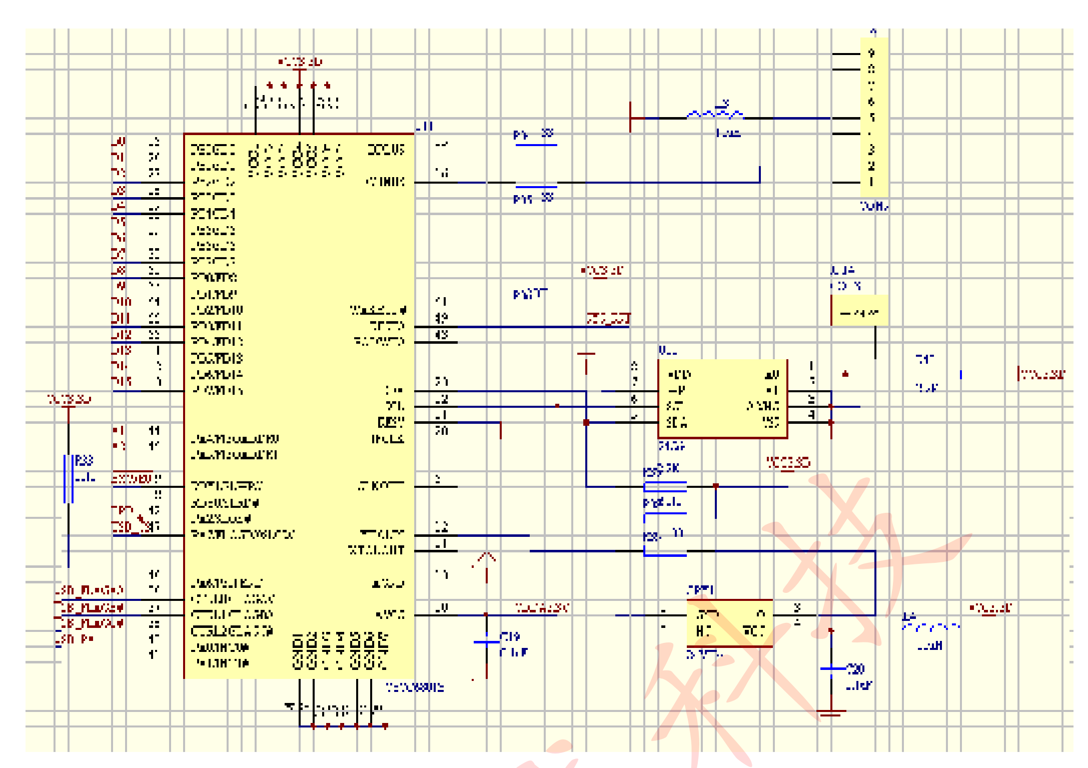
引脚分配如下图所示：



9. USB 通信接口单元

CPU 板采用的是 CYPRESS 公司推出的 USB2.0 控制器 CY7C68013 芯片,该芯片是从控制芯片,符合 USB2.0 规范,并能向下兼容 USB1.1 协议.支持全速(12M),低速(1.5M)和高速(480M)三种传输速率.(更加具体芯片介绍请参看其数据手册)

接口电路如下：



1. USB 片选地址:

USB_FIF02	0x190000	用于接收从 PC 机发送来的数据
USB_FIF04	0x190002	
USB_FIF06	0x190004	用于将数据返回给 PC 机
USB_FIF08	0x190006	

2.定义状态寄存器

为了在不使用中断时，仍然可以使用 USB 与 PC 机进行数据通信。在 CPLD 内部定义了一个状态寄存器，用于监控 USB 控制芯片（68013）的 FIFO 状态.

CPLD 内部自定义寄存器地址：0x1a0000（只读）

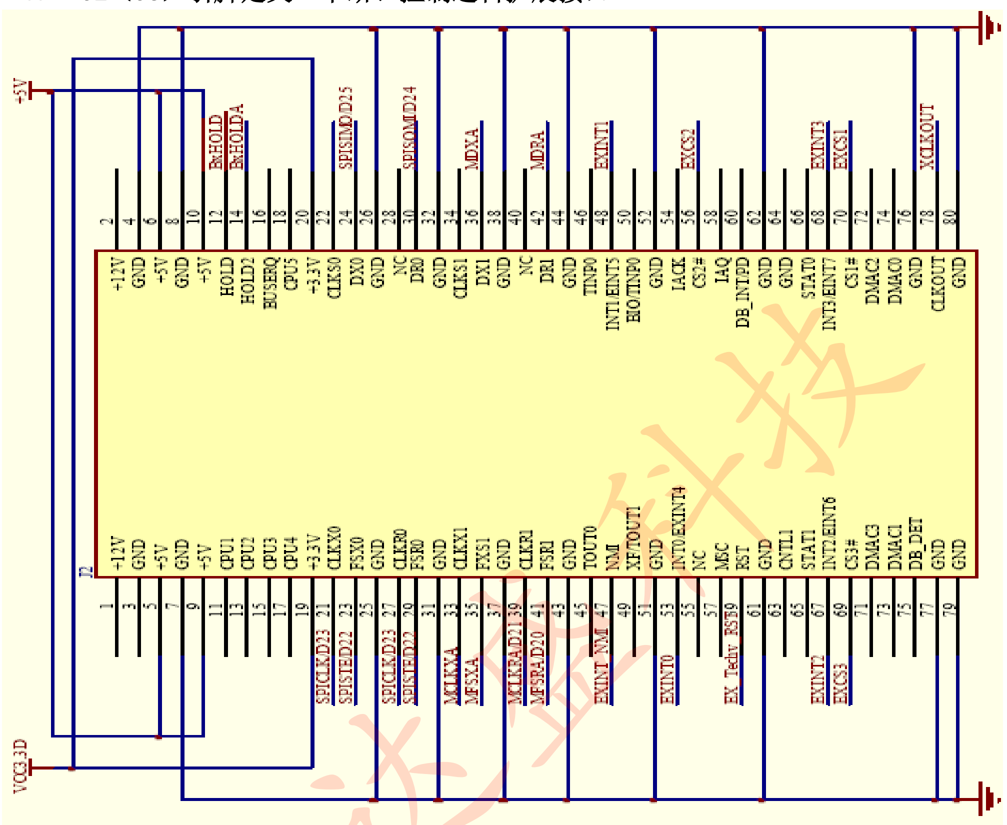
3	2	1	0
PA	USB_FLAGC	USB_FLAGB	USB_FLAGA

68013 的状态引脚定义:

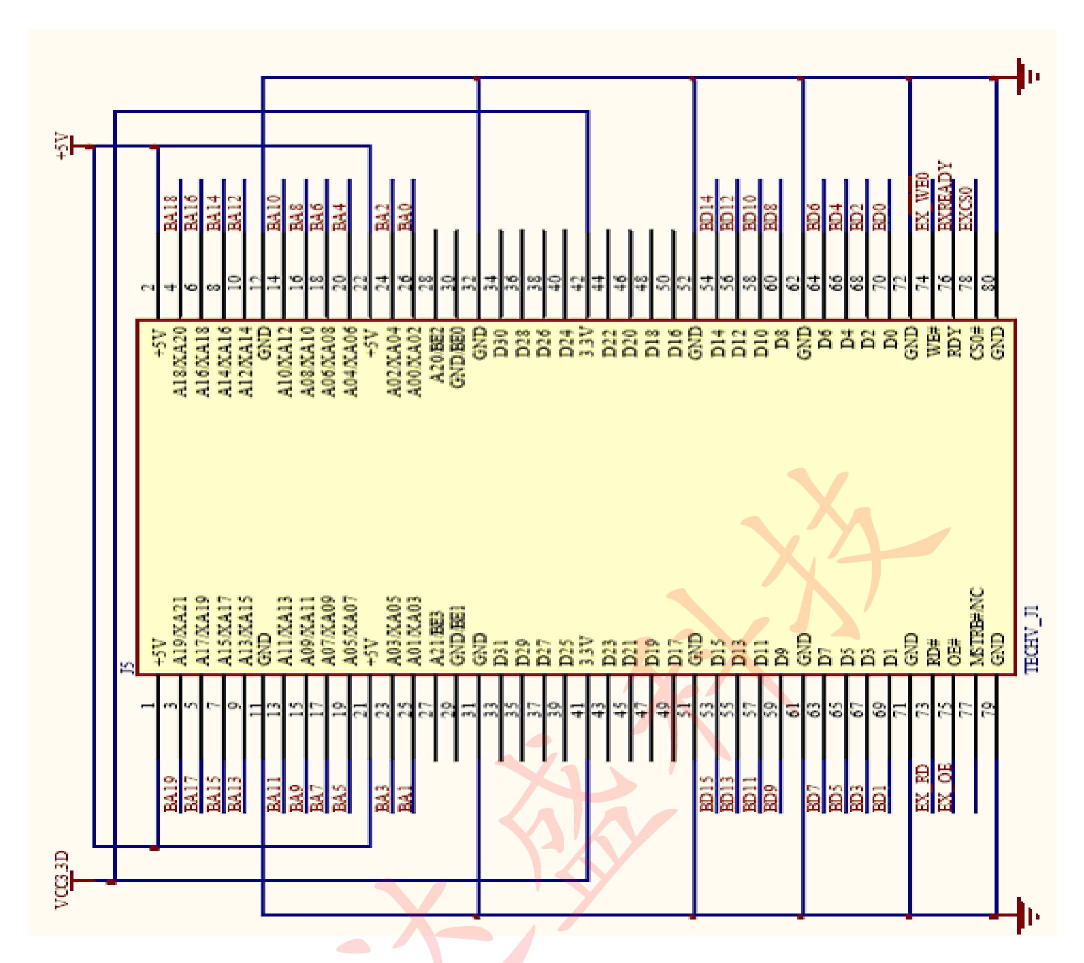
FLAGA	68013 的 FIF02 的数据多于 64 个字节时变为低，否则为高。
USB_FLAGB	FIF06 为空时输出低，否则为高。
USB_FLAGC	68013 的 FIF02 空时为低，否则为高。
PA	监控 68013 的 portA0 引脚，暂时保留使用。

3.扩展接口介绍

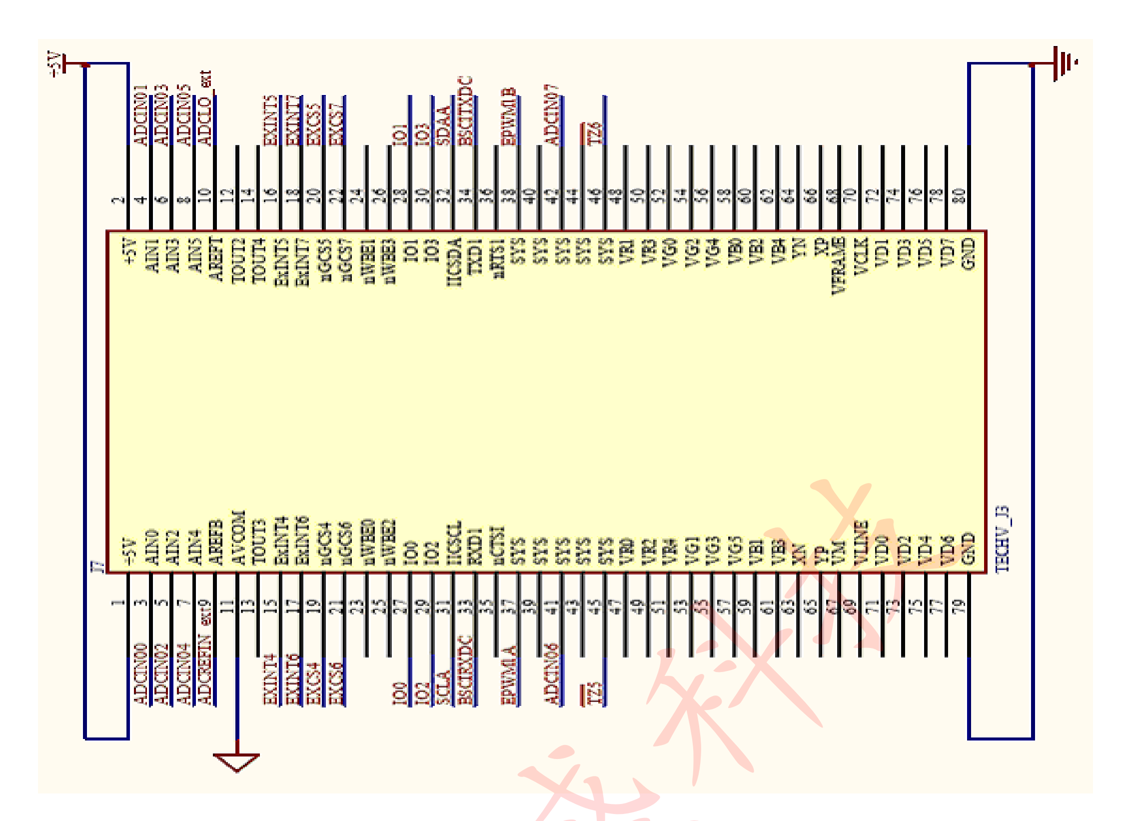
A. J2（J3）引脚定义—中断、控制逻辑扩展接口



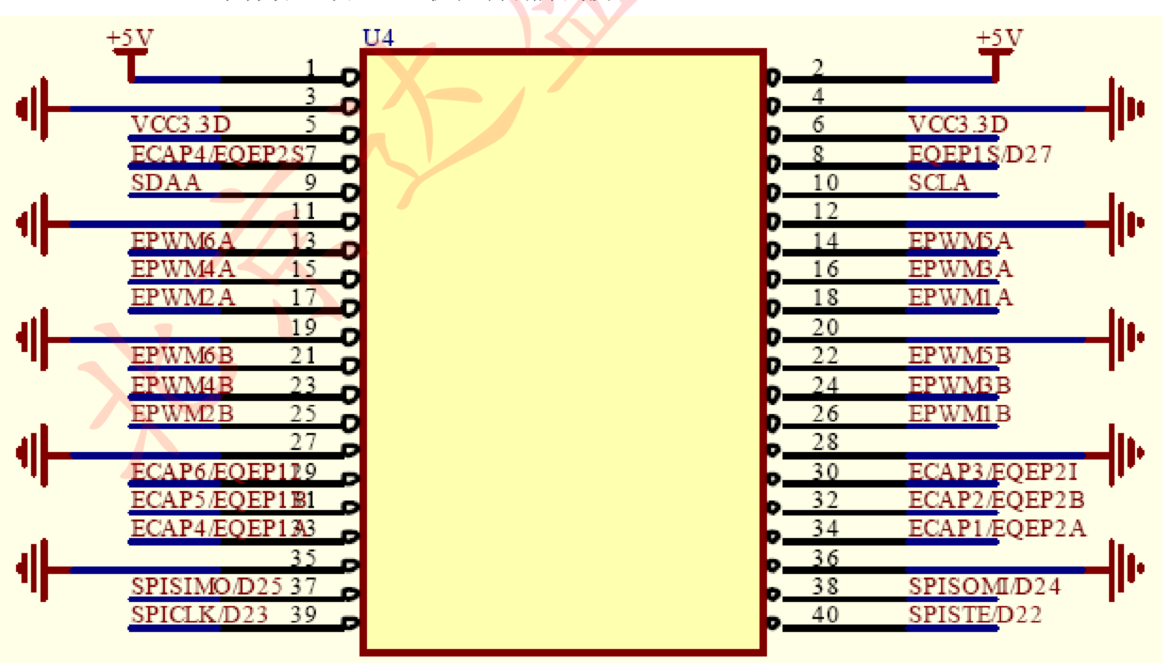
B. J4(J5)引脚定义—扩展总线接口



C. J6（J7）引脚定义—AD、事件管理单元、



D. U4--SPI、事件管理单元、电机控制引脚线接口



- J2，J4，J6，U4 引脚具体定义说明如下：

注意：下表中“F28335 用到的管脚含义”一栏中是 TMS320F28335 板子中用到的引脚，其他未用到的管脚用“NC”来表示。

“管脚名”是通用扩展总线中的引脚名称，请阅读时请注意区分。

- **J2 定义说明：**

管脚号	管脚名	F28335 用到的管脚含义	IO	备注
1	+12V	NC		+12V 电源
2	+12V	NC		+12V 电源
3	GND	地		地
4	GND	地		地
5	+5V	+5V 电源		+5V 电源
6	+5V	+5V 电源		+5V 电源
7	GND	地		地
8	GND	地		地
9	+5V	+5V 电源		+5V 电源
10	+5V	+5V 电源		+5V 电源
11	CPU1	NC	O	CPU 种类指示信号
12	HOLD	BXHOLD	I	外部总线保持信号
13	CPU2	NC	O	CPU 种类指示信号
14	HOLDA	BxHOLDA	O	总线保持响应信号
15	CPU3	NC	O	CPU 种类指示信号
16	BUSREQ	NC	O	外部总线请求信号
17	CPU4	NC	O	CPU 种类指示信号
18	CPU5	CPU 种类指示信号	O	CPU 种类指示信号
19	+3.3V	+3.3V 电源		+3.3V 电源
20	+3.3V	+3.3V 电源		+3.3V 电源
21	CLKX0	SPICLK/D23	IO	McBSP0 输出位时钟
22	CLKS0	NC	I	McBSP0 外部输入时钟
23	FSX0	SPISTE/D22	IO	McBSP0 输出帧时钟
24	DX0	SPISIMO/D25	O	McBSP0 输出数据
25	GND	地		地
26	GND	地		地
27	CLKR0	SPICLK/D23	IO	McBSP0 输入位时钟

28	NC	NC		NC
29	FSR0	SPISTE/D22	IO	McBSP0 输入帧时钟
30	DR0	SPISOMI/D24	I	McBSP0 输入数据
31	GND	地		地
32	GND	地		地
33	CLKX1	MCLKXA	IO	McBSP1 输出位时钟
34	CLKS1	NC	I	McBSP1 外部输入时钟
35	FSX1	MFSXA	IO	McBSP1 输出帧时钟
36	DX1	MDXA	O	McBSP1 输出数据
37	GND	地		地
38	GND	地		地
39	CLKR1	MCLKRA/D21	IO	McBSP1 输入位时钟
40	NC	NC		空脚
41	FSR1	MFSRA/D20	IO	McBSP1 输入帧时钟
42	DR1	MDRA	I	McBSP1 输入数据
43	GND	地		地
44	GND	地		地
45	TOUT0	TAPWM/TACMP	O	定时器输出 0
46	TINP0	NC	I	定时器输入 0
47	NMI	EXINT_NMI	I	不可屏蔽中断
48	INT1/EINT5	EXINT1	I	中断
49	XF/TOUT1	NC	O	O/定时器 1 输出
50	BIO/TINP1	NC	I	IN/定时器 1 输入
51	GND	地		地
52	GND	地		地
53	INT0/EINT4	EXINT0	I	中断
54	IACK	NC	O	中断响应信号
55	NC	NC		空脚

56	CS2	EXCS2	O	片选信号 2
57	MSC	NC	O	状态完成信号
58	IAQ	NC	O	地址采集信号
59	RESET	EX_ Techv_RST	O	复位信号
60	DBINT/PD	NC		子板中断/电源指示
61	GND	地		地
62	GND	地		地
63	CNTL1	NC	O	子板控制信号 1
64	CNTL0	NC	O	子板控制信号 0
65	STAT1	NC	I	子板状态信号 1
66	STAT0	NC	I	子板状态信号 0
67	INT2/EINT6	EXINT2	I	中断
68	INT3/EINT7	EXINT3	I	中断
69	CS3	EXCS3	O	片选信号 3
70	CS1	EXCS1	O	片选信号 1
71	DMAC3	NC	O	DMA 状态信号
72	DMAC2	NC	O	DMA 状态信号
73	DMAC1	NC	O	DMA 状态信号
74	DMAC0	NC	O	DMA 状态信号
75	DB_DET	NC	I	子板检测信号
76	GND	地		地
77	GND	地		地
78	CLKOUT	XCLKOUT	O	时钟
79	GND	地		地
80	GND	地		地

- J4 引脚定义说明：

管脚号	管脚名	F28335 用到的管脚含义	IO	备注
1	+5V	+5V 电源		+5V 电源
2	+5V	+5V 电源		+5V 电源

3	A19|XA21	BA19	O	地址线
4	A18|XA20	BA18	O	地址线
5	A17|XA19	BA17	O	地址线
6	A16|XA18	BA16	O	地址线
7	A15|XA17	BA15	O	地址线
8	A14|XA16	BA14	O	地址线
9	A13|XA15	BA13	O	地址线
10	A12|XA14	BA12	O	地址线
11	GND	地		地
12	GND	地		地
13	A11|XA13	BA11	O	地址线
14	A10|XA12	BA10	O	地址线
15	A09|XA11	BA9	O	地址线
16	A08|XA10	BA8	O	地址线
17	A07|XA09	BA7	O	地址线
18	A06|XA08	BA6	O	地址线
19	A05|XA07	BA5	O	地址线
20	A04|XA06	BA4	O	地址线
21	+5V	+5V 电源		+5V 电源
22	+5V	+5V 电源		+5V 电源
23	A03|XA05	BA3	O	地址线
24	A02|XA04	BA2	O	地址线
25	A01|XA03	BA1	O	地址线
26	A00|XA02	BA0	O	地址线
27	A21|BE3	NC	O	地址线
28	A20|BE2	NC	O	地址线
29	NC|BE1	NC	O	地址线
30	NC|BE0	NC	O	地址线

31	GND	地		地
32	GND	地		地
33	D31	NC	IO	数据线
34	D30	NC	IO	数据线
35	D29	NC	IO	数据线
36	D28	NC	IO	数据线
37	D27	NC	IO	数据线
38	D26	NC	IO	数据线
39	D25	NC	IO	数据线
40	D24	NC	IO	数据线
41	+3.3V	+3.3V 电源		+3.3V 电源
42	+3.3V	+3.3V 电源		+3.3V 电源
43	D23	NC	IO	数据线
44	D22	NC	IO	数据线
45	D21	NC	IO	数据线
46	D20	NC	IO	数据线
47	D19	NC	IO	数据线
48	D18	NC	IO	数据线
49	D17	NC	IO	数据线
50	D16	NC	IO	数据线
51	GND	地		地
52	GND	地		地
53	D15	BD15	IO	数据线
54	D14	BD14	IO	数据线
55	D13	BD13	IO	数据线
56	D12	BD12	IO	数据线
57	D11	BD11	IO	数据线
58	D10	BD10	IO	数据线

59	D9	BD9	IO	数据线
60	D8	BD8	IO	数据线
61	GND	地		地
62	GND	地		地
63	D7	BD7	IO	数据线
64	D6	BD6	IO	数据线
65	D5	BD5	IO	数据线
66	D4	BD4	IO	数据线
67	D3	BD3	IO	数据线
68	D2	BD2	IO	数据线
69	D1	BD1	IO	数据线
70	D0	BD0	IO	数据线
71	GND	地		地
72	GND	地		地
73	RD	EX_RD	O	读信号
74	WE	EX_WE0	O	写信号
75	OE	EX_OE	O	使能信号
76	RDY	BXREADY	I	准备好信号
77	MSTRB	NC	O	存储器选通信号
78	CS0	EXCS0	O	片选信号 0
79	GND	地		地
80	GND	地		地

l J6 定义说明

管脚号	管脚名	F28335 用到的管脚含义	IO	备注
1	+5V	+5V 电源		
2	+5V	+5V 电源		
3	AIN0	ADCIN00	I	DSP-TMS320F28335 板子芯片 ADCINA0 和 ADCINA1 引出脚

4	AIN1	ADCIN01	I	DSP-TMS320F28335 板子芯片 ADCINA2 和 ADCINA3 引出脚
5	AIN2	ADCIN02	I	DSP-TMS320F28335 板子芯片 ADCIN4 和 ADCINA5 引出脚
6	AIN3	ADCIN03	I	DSP-TMS320F28335 板子芯片 ADCINA6 和 ADCINA7 引出脚
7	AIN4	ADCIN04	I	DSP-TMS320F28335 板子芯片 ADCINB0 和 ADCINB1 引出脚
8	AIN5	ADCIN05	I	DSP-TMS320F28335 板子芯片 ADCINB2 和 ADCINB3 引出脚
9	AREFB	ADCREFIN_ext	I	DSP-TMS320F28335 板子芯片 ADC 模块外部参考电压输入
10	AREFT	ADCLO_ext	I	DSP-TMS320F28335 板子芯片 ADC 模块外部参考电压
11	AVCOM	NC		Techv 扩展板通用管脚
12	TOUT2	NC	O	GP Timer2
13	TOUT3	NC	O	GP Timer3
14	TOUT4	NC	O	GP Timer4
15	ExINT4	EXINT4	IO	
16	ExINT5	EXINT5	IO	
17	ExINT6	EXINT6	IO	
18	ExINT7	EXINT7	IO	
19	nGCS4	EXCS4	IO	

20	nGCS5	EXCS5	IO	
21	nGCS6	EXCS6	IO	
22	nGCS7	EXCS7	IO	
23	nWBE0	NC		Techv 扩展板通用管脚
24	nWBE1	NC		Techv 扩展板通用管脚
25	nWBE2	NC		Techv 扩展板通用管脚
26	nWBE3	NC		Techv 扩展板通用管脚
27	IO0	IO0	IO	连接到 CPLD 管脚
28	IO1	IO1	IO	连接到 CPLD 管脚
29	IO2	IO2	IO	连接到 CPLD 管脚
30	IO3	IO3	IO	连接到 CPLD 管脚
31	IICSCL	SCLA		串行时钟总线
32	IICSDA	SDAA		串行数据总线
33	RXD1	BSCIRXDC		Techv 扩展板通用管脚
34	TXD1	BSCITXDC		Techv 扩展板通用管脚
35	nCTS1	NC		Techv 扩展板通用管脚
36	nRTS1	NC		Techv 扩展板通用管脚
37	SDDAT0	EPWM1A		Techv 扩展板通用管脚
38	SDDAT1	EPWM1B		Techv 扩展板通用管脚
39	SDDAT2	NC		Techv 扩展板通用管脚
40	SDDAT3	NC		Techv 扩展板通用管脚
41	SDCLK	ADCIN06		DSP-TMS320F28335 板子芯片 ADCINB4 和 ADCINB5 引出脚
42	SDCMD	ADCIN07		DSP-TMS320F28335 板芯片 ADCINB6 和 ADCINB7 引出脚
43	SYS	NC		Techv 扩展板通用管脚
44	SYS	NC		Techv 扩展板通用管脚

45	SYS	TZ5		Techv 扩展板通用管脚
46	CODCLK	TZ6		Techv 扩展板通用管脚
47	VR0	NC		Techv 扩展板通用管脚
48	VR1	NC		Techv 扩展板通用管脚
49	VR2	NC		Techv 扩展板通用管脚
50	VR3	NC		Techv 扩展板通用管脚
51	VR4	NC		Techv 扩展板通用管脚
52	VG0	NC		Techv 扩展板通用管脚
53	VG1	NC		Techv 扩展板通用管脚
54	VG2	NC		Techv 扩展板通用管脚
55	VG3	NC		Techv 扩展板通用管脚
56	VG4	NC		Techv 扩展板通用管脚
57	VG5	NC		Techv 扩展板通用管脚
58	VB0	NC		Techv 扩展板通用管脚
59	VB1	NC		Techv 扩展板通用管脚
60	VB2	NC		Techv 扩展板通用管脚
61	VB3	NC		Techv 扩展板通用管脚
62	VB4	NC		Techv 扩展板通用管脚
63	XN	NC		Techv 扩展板通用管脚
64	YN	NC		Techv 扩展板通用管脚
65	YP	NC		Techv 扩展板通用管脚
66	XP	NC		Techv 扩展板通用管脚
67	VM	NC		Techv 扩展板通用管脚
68	VFRAME	NC		Techv 扩展板通用管脚
69	VLINE	NC		Techv 扩展板通用管脚
70	VCLK	NC		Techv 扩展板通用管脚
71	VD0	NC		Techv 扩展板通用管脚
72	VD1	NC		Techv 扩展板通用管脚

73	VD2	NC		Techv 扩展板通用管脚
74	VD3	NC		Techv 扩展板通用管脚
75	VD4	NC		Techv 扩展板通用管脚
76	VD5	NC		Techv 扩展板通用管脚
77	VD6	NC		Techv 扩展板通用管脚
78	VD7	NC		Techv 扩展板通用管脚
79	GND	GND		GND
80	GND	GND		GND

- U4 引脚定义说明(DSP 芯片部分引脚线)

注意：这些引脚直接和 F28335 芯片引脚相连接，因此，这些引脚为 TTL3.3V 标准，其输出最高电压为 3.3V，如果要接入 5V 器件，则需要外接电平转换电路。

管脚序号	代号	含义	备注
1	+5V	+5V 电源	
2	+5V	+5V 电源	
3	GND	接地	
4	GND	接地	
5	VCC3.3D	+3.3V	
6	VCC3.3D	+3.3V	
7	ECAP4/EQEP2S		
8	EQEP1S/D27		
9	SDAA	IIC 数据总线	
10	SCLA	IIC 时钟总线	
11	GND	接地	
12	GND	接地	
13	EPWM6A	EPWM6A 输出引脚	
14	EPWM5A	EPWM5A 输出引脚	
15	EPWM4A	EPWM4A 输出引脚	
16	EPWM3A	EPWM3A 输出引脚	
17	EPWM2A	EPWM2A 输出引脚	

18	EPWM1A	EPWM1A 输出引脚	
19	GND	接地	
20	GND	接地	
21	EPWM6B	EPWM6B 输出引脚	
22	EPWM5B	EPWM5B 输出引脚	
23	EPWM4B	EPWM4B 输出引脚	
24	EPWM3B	EPWM3B 输出引脚	
25	EPWM2B	EPWM2B 输出引脚	
26	EPWM1B	EPWM1B 输出引脚	
27	GND	接地	
28	GND	接地	
29	ECAP6/EQEP1I	ECAP6/EQEP1I 输出引脚	
30	ECAP3/EQEP2I	ECAP3/EQEP2I 输出引脚	
31	ECAP5/EQEP1B	ECAP5/EQEP1B 输出引脚	
32	ECAP2/EQEP2B	ECAP2/EQEP2B 输出引脚	
33	ECAP4/EQEP1A	ECAP4/EQEP1A 输出引脚	
34	ECAP1/EQEP2A	ECAP1/EQEP2A 输出引脚	
35	GND	接地	
36	GND	接地	
37	SPISIMO/D25	SPISIMO 引出脚	
38	SPISOMI/D24	SPISOMI 引出脚	
39	SPICLK/D23	SPICLK 输出脚	
40	SPISTE/D22	SPISTE 引出脚	

说明:

上述 J2,J3,J4,J5 管脚接口是 Tech_v 总线接口,该接口是和 TI 公司 DSK 兼容的信号扩展接口,可连接我公司的图象处理、高速 AD、DA、USB 和以太网等扩展板,也可以连接 TI 公司的标准 DSK 扩展信号板.

第二章软件环境的搭建

1. 安装调试软件

如果用户使用 TDS510USB 仿真器请安装 CCS3.2，如果是 XDS510USB 请安装 CCS3.3。

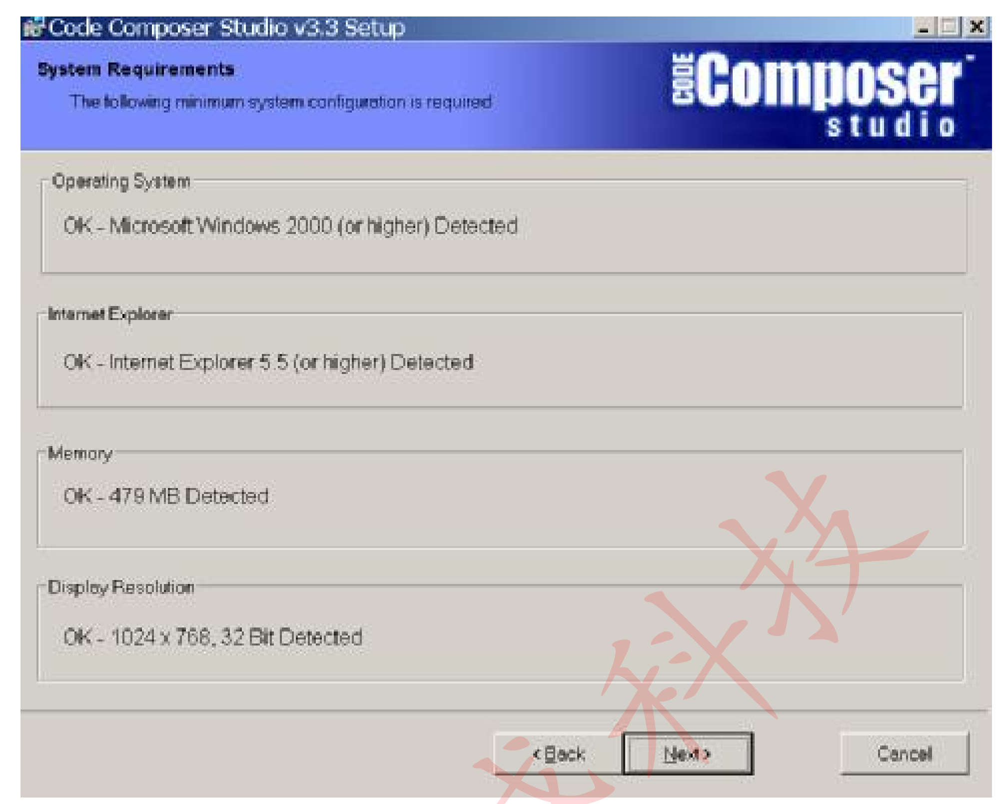
.1.1 安装 DSP 调试软件－CCS3.3



1) 双击目录 CCS3.3 下的 setup.exe 开始安装，出现如下界面：



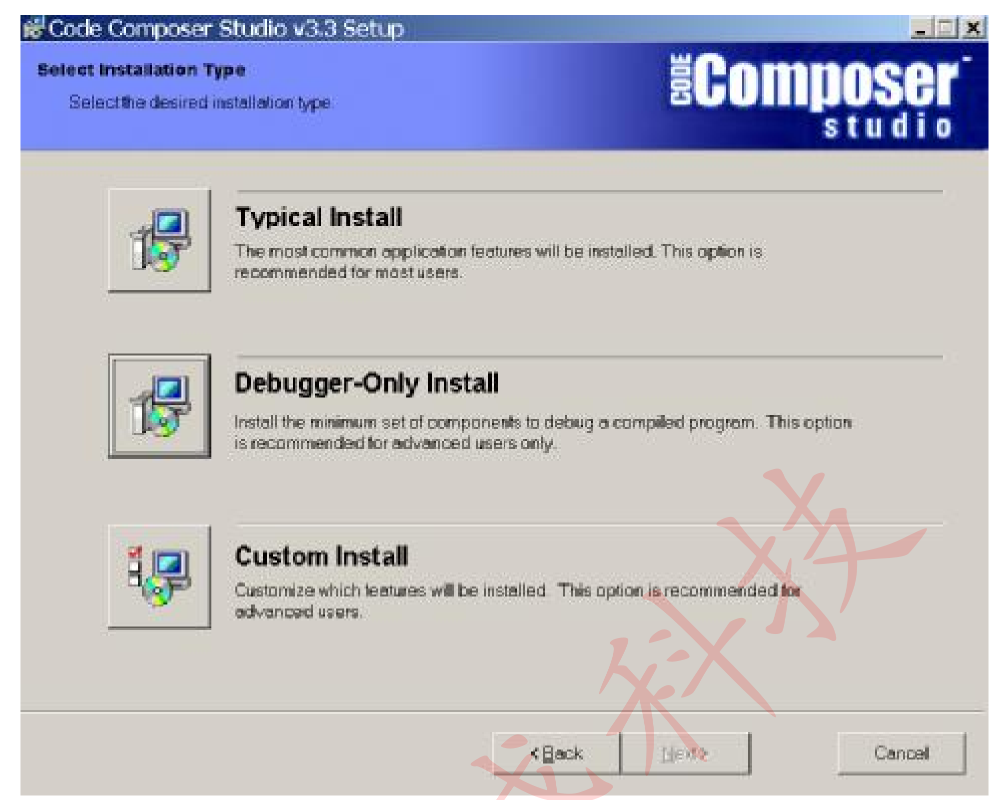
2) 点 next，检测 OK



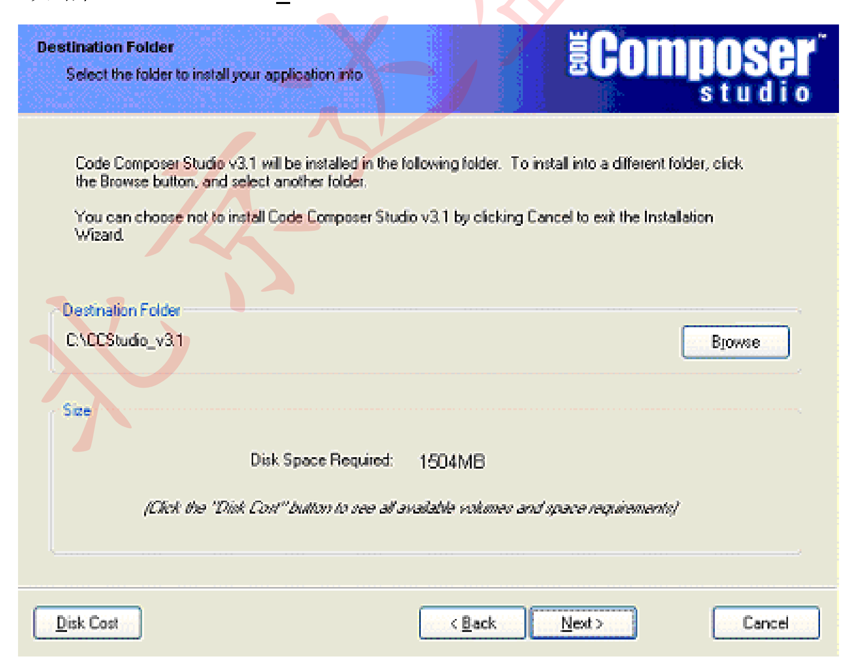
3) 点 next，询问 license



4) 点同意，next，询问安装方式

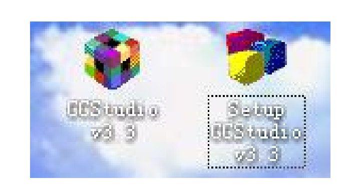


5) 我们选择自定义安装，即“Typical Install”，在出现下面的界面后，建议按装在默认路径“C:\CCStudio_v3.3”



点击“Next”后，进行安装

6) 点击“Next”,进行安装，安装完后出现下面的图标



1.2 CCS3.3软件升级包的安装：

在 CCS3.3 升级包中含有：C2000CodeGenerationTools5[1].0.0Beta2.exe ；CCS_v3.3_SR11_81.6.2.exe；F2823x_RevA_CSP.exe；三个文件。

1. 双击"CCS_v3.3_SR11_81.6.2.exe"，一路狂点"NEXT"，最后该软件会安装在 CCS3.3 软件的默认安装路径下(建议安装在 C:\CCStudio_v3.3)。
2. 双击"C2000CodeGenerationTools5[1].0.0Beta2.exe"，一路狂点"NEXT"，最后该软件会安装在 CCS3.3 软件的默认安装路径下（建议安装在 C:\CCStudio_v3.3）。
3. 双击"F2823x_RevA_CSP.exe"，一路狂点"NEXT"，最后该软件会安装在 CCS3.3 软件的默认安装路径下（建议安装在 C:\CCStudio_v3.3）。

此时三个升级软件安装完成。

1.3安装DSP仿真器－XDS510 USB驱动

该型号仿真器接口为 USB，所以驱动也分为两个，一个是电脑的 Windows 驱动，另一为 CCS 驱动，下面分别叙述。

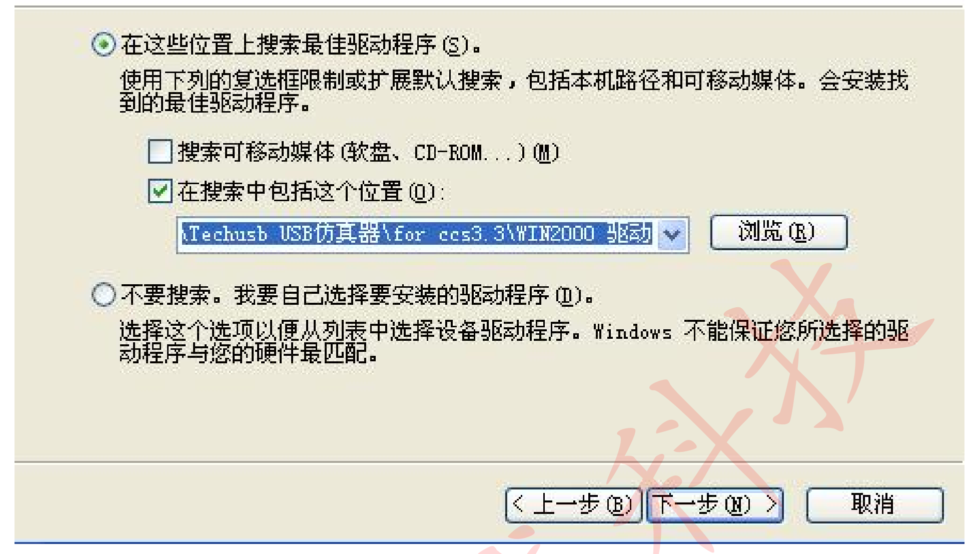
1) 安装 Windows 驱动

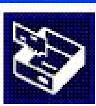
将电脑和 USB 仿真器连接起来，会出现找到新的硬件向导如下：



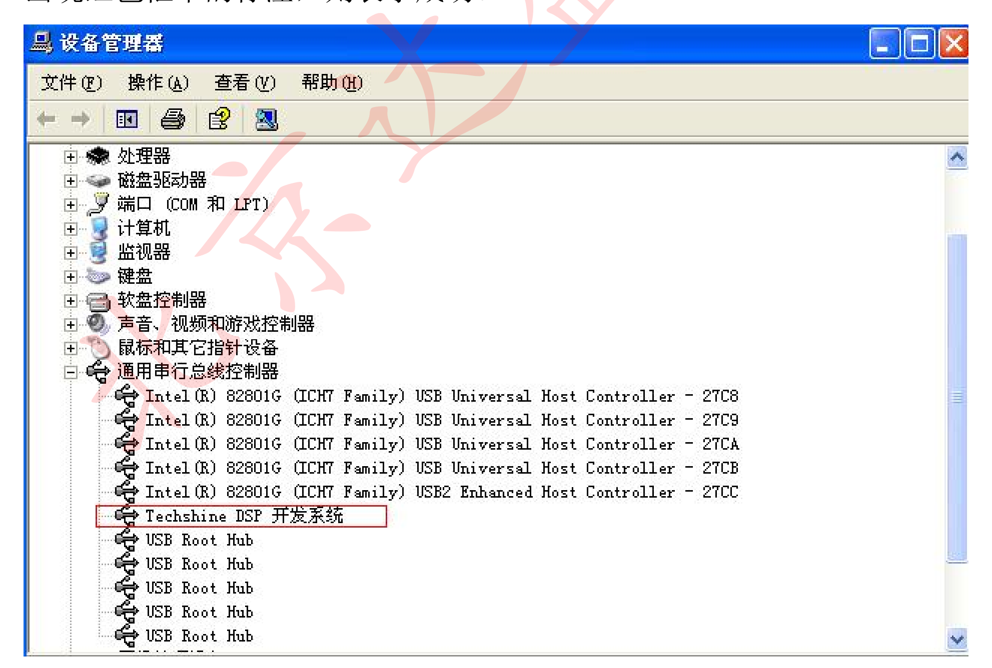
点击"从列表或指定位置安装（高级）"，然后点击"下一步"，在下面的界面中找到"WIN2000 驱动"所在的位置，（例：--\仿真器驱动\USB 仿真器驱动\510 仿真器驱动\Techusb USB 仿真器\for ccs3.3\WIN2000 驱动）

然后点击"NEXT"，此刻则安装成功。





点击“我的电脑”—》“属性”—》“硬件”—》“设备管理器”：
出现红色框中的标注，则表示成功。



2) 安装 CCS3.3 驱动

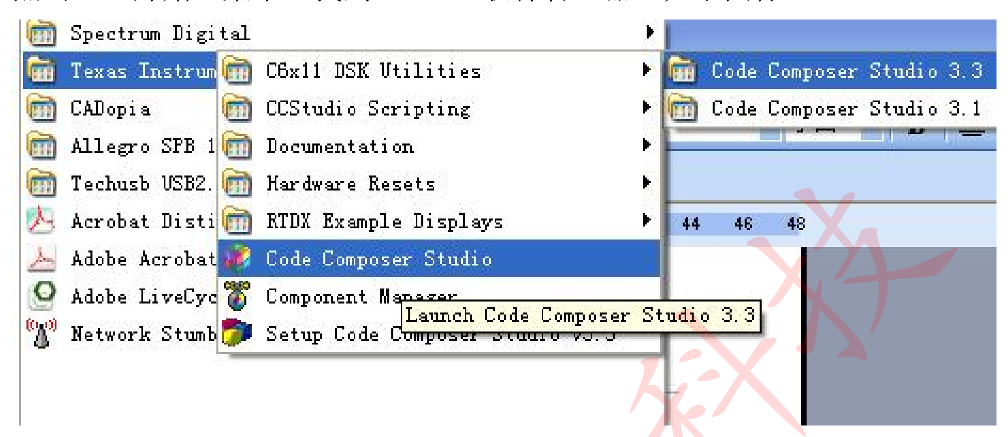
安装文件位于目录“software\XDS510USB_CCS3.3_drvier”下。双击图标“TechUSB_For_CCS_3.3_Setup.exe”开始安装：
一路点击“NEXT”，注意安装目录要和 CCS3.3 软件一致。

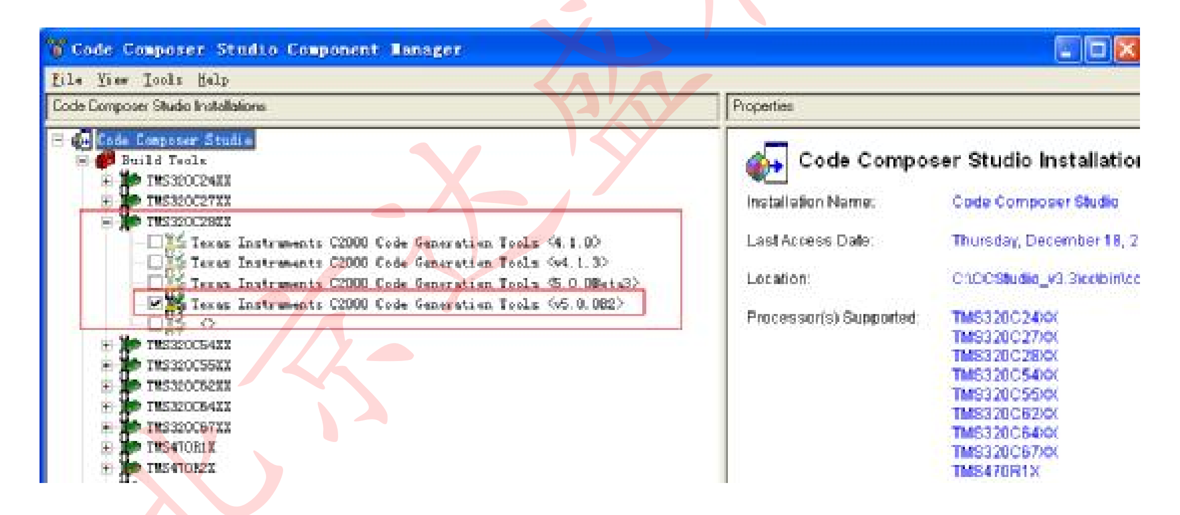
至此 CCS3.3 的驱动也已经安装完毕！

2. 配置XDS510 USB(使用CCS3.3)

2.1 更改设置管理器设置

点击 “ 开始”菜单，找到 CCS3.3 软件管理器，如下图标：

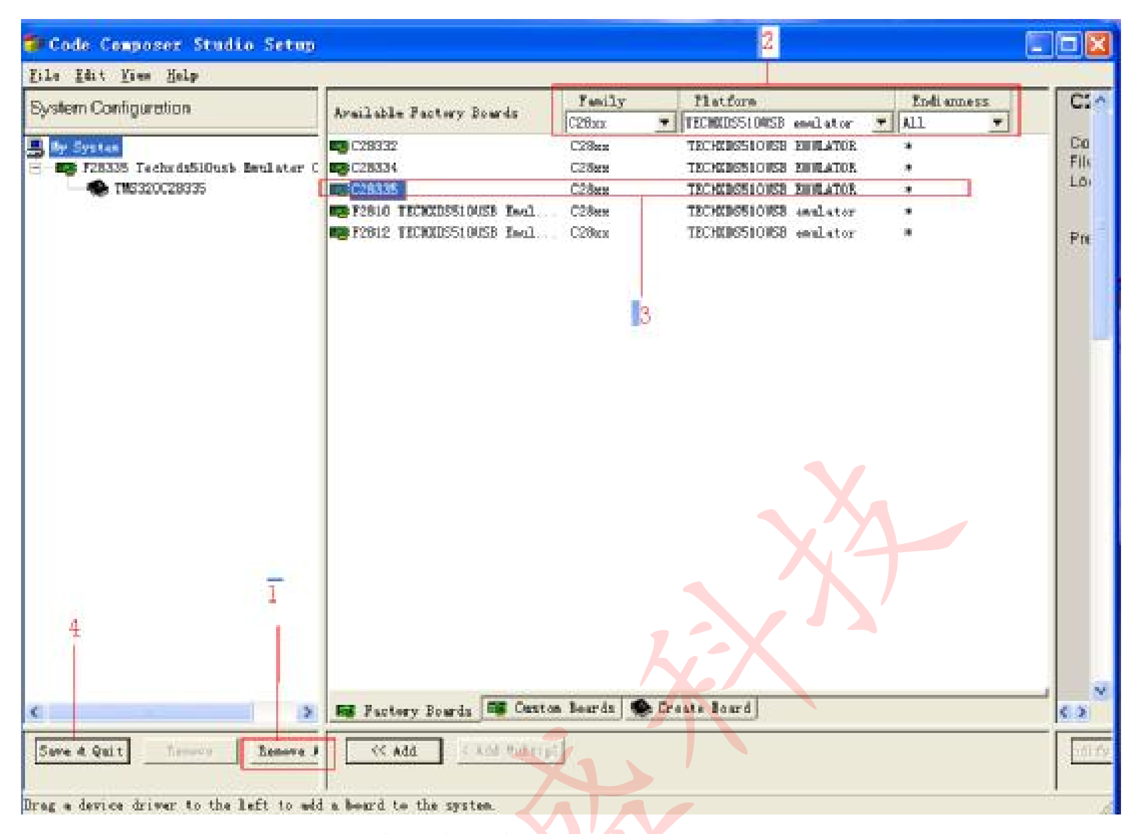




选中上面大红框中的小红框内容：Texas Instruments C2000 Code Generation Tools<v5.0.0b2>
关闭保存退出。

2.2 CCS3.3软件的设置

打开 CCS3.3 的设置程序 Setup CCStudio v3.3，出现如下界面：



先点击“1”，然后再在“2”中三个框中依次选择如上图，然后双击“3”，然后在点击“4”，出现如下界面



点击“是（Y)”，下面的则是进入 CCS3.3 软件内进行仿真实验了。

第三章 CPU 板测试程序

实验一 LED 实验

1：实验目的：

a 了解 DSP 开发系统的组成和结构

b 了解 Techv_28335 上自定义 LED 灯的编程使用方法

2：实验器件

计算机、TechV28335 板、仿真器(510USB2.0PLUS)、CCS3.3 以上版本。

3：实验内容：

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

4：实验步骤

1、运行 Code Composer Studio (CCS3.3)；

2、打开系统项目文件 \......\ CPU board\1 Led Test\example.pjt；

3、编译全部文件并装载；

4、跳到主程序的开始；

5、全速运行程序，可以看到板卡上的 LED5—LED8 循环点亮。

5:试验结束

实验二 CPU 定时实验

1：实验目的：

a 了解 DSP 开发系统的组成和结构；

b 掌握 F28335 的 CPU 定时器的控制方法；

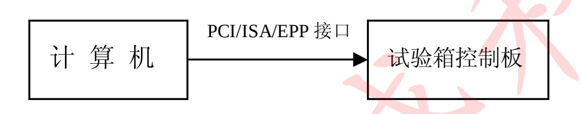
2：实验器件

计算机、TechV28335CPU 板、仿真器(510USB2.0PLUS)、CCS3.3 以上版本。

3：实验内容：

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

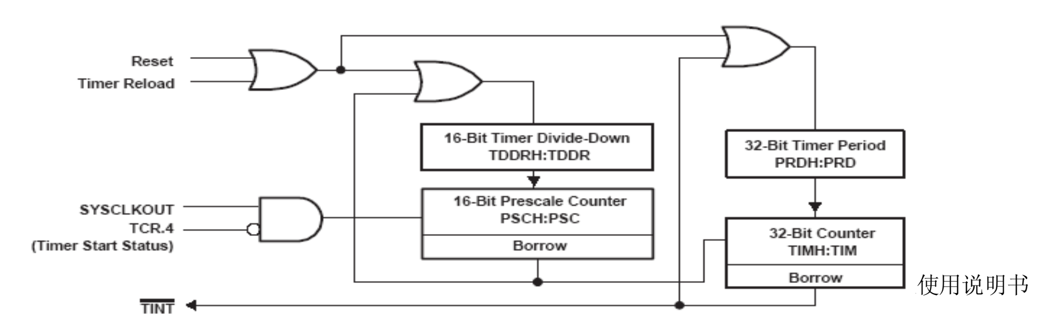
4.：试验原理：

样例实验是采用 CPU 定时器来定时使 LED 亮灭的。

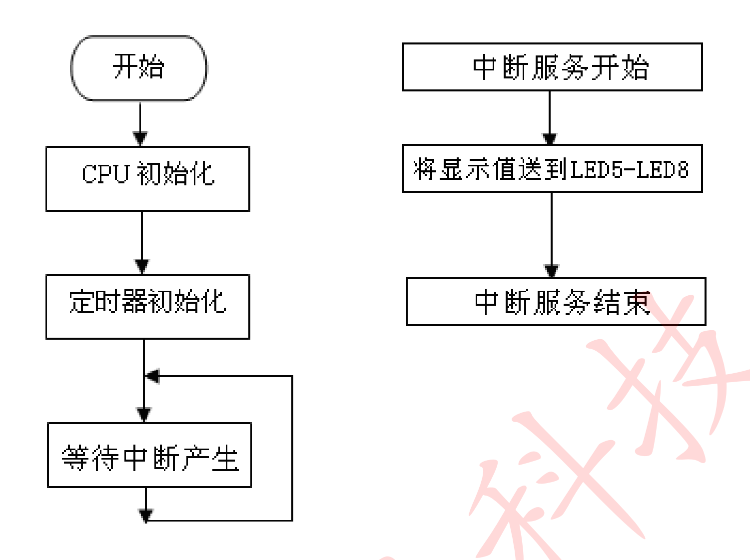
F28335 的 CPU 共有三个定时器，其中，CPU 定时器 2 被保留用作实时操作系统 OS(例如 DSPBIOS)，CPU 定时器 0 和定时器 1 可以供用户使用。

定时器的一般操作如下：将周期寄存器 PRDH:PRD 中的值装入 32 为计数器寄存器 TIMH:TIM。然后计数器寄存器以 F28XX 的 SYSCLKOUT 速率递减。当计数器减到 0 时，就会产生一个定时器中断输出信号（一个中断脉冲）。（更加详细的原理请参看 TI DATESHEET）

下图为 CPU 定时器的内部结构：



样例实验的程序框图如下：



5：实验步骤

1、运行 Code Composer Studio (CCS3.3)；

2、打开系统项目文件 \......\ CPU board\2 Cputimer\ example.pjt；

3、编译全部文件并装载；

4、单击 "DEBUG\GO MAIN"跳到主程序的开始；

5、单击 "DEBUG\Animate"全速运行程序， 可以观察到 CPU 板上的 LED5-LED8 灯以一定的定时周期做闪烁变化.

6：实验结束

实验三 SRAM 测试实验

1：实验目的：

a 了解 DSP 开发系统的组成和结构；

b 了解 SRAM 的配置和读写操作；

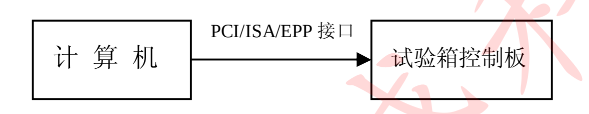
2：实验器件

计算机、TechV28335CPU 板、仿真器(510USB2.0PLUS)、CCS3.3 以上版本。

3：实验内容：

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

4:**实验原理**:

本样例实验是通过对 CPU 扳上的 SRAM,进行存储空间段的数据写入,然后再读出所写入的数据,两者进行一致性校对,如果校对一致性,则 CPU 扳上的 LED5-LED8 灯会依次闪烁变化,反之,则 LED6 灯和 LED8 灯点亮,且程序停止运行.

SRAM 的存储空间单元段:0x100000—0x10ffff(64K)

4：实验步骤

1、运行 Code Composer Studio (CCS)；

2、打开系统项目文件 \......\ CPU board\3 SRAM Test\ example.pjt；

3、编译全部文件并装载；

4、跳到主程序的开始；

5、全速运行程序，如果板卡上的 LED5—LED8 亮灭变化，则测试通过；如果 LED6 灯和 LED8 灯点亮,且程序停止运行.则测试不通过；

实验四 URAT 测试实验

一、 实验目的

1． 学习 RS232 串口通讯原理；

2． 学习‘C2000 系列 DSP 的 RS232 串行口模块的操作方法。

二、 实验设备

计算机，CCS 3.3 版软件， DSP 仿真器 ，F28335CPU 板

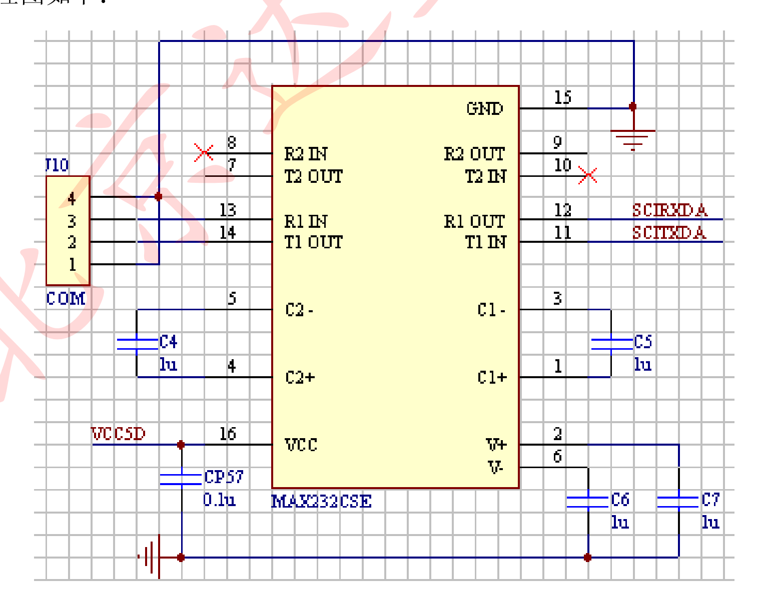
三、 实验原理

串行通信接口 SCI 是一个双线的异步串口，一般看作是 UART。SCI 模块支持 CPU 与采样非返回至 0（NRZ）标准格式的异步外围设备之间的数字通信。SCI 的接收器和发送器各具有一个 16 级深度的 FIFO，这样可以减少空头的服务。它们还各有自己独立的使能位和中断位，可以在半双工通信中进行独立的操作，或在全双工通信中同时进行操作。

为了确保数据的完整性，SCI 检查所接收数据的中断检测、极性、溢出和帧错误。位速率可通过编程一个 16 位的波特率改变寄存器而改变。

F28335 芯片共有三个 SCI 模块：即 SCIA，SCIB 和 SCIC。每个 SCI 模块都有两个外部引脚：SCITXD 和 SCIRXD；波特率可通过编程达到 64K 不同的速率；半双工或全双工的工作模式；编程的数据长度 8～16 位可供用户选择。

本实验模块原理图如下：



详细的用法请参看例程

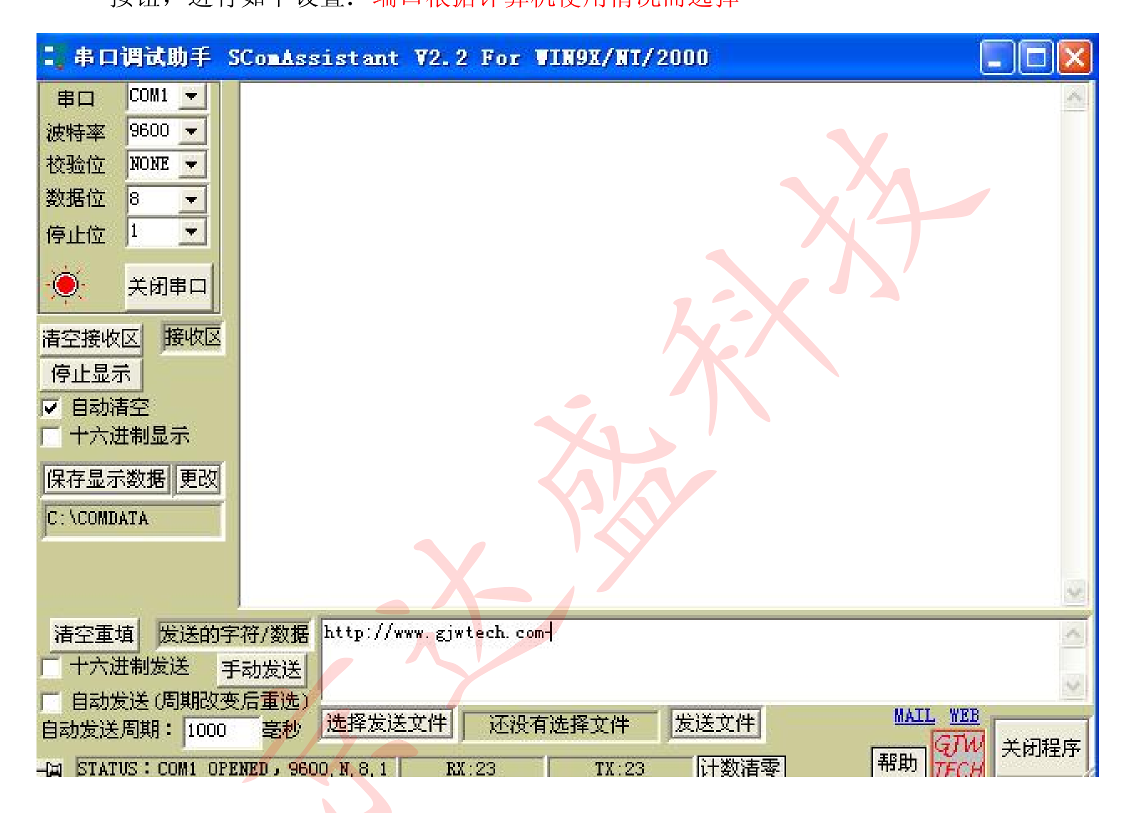
四、 实验步骤

1． 将实验箱所配的**直连串口线的一端**连接电脑的串口 1,另一端(近似网口)连接到 28335CPU

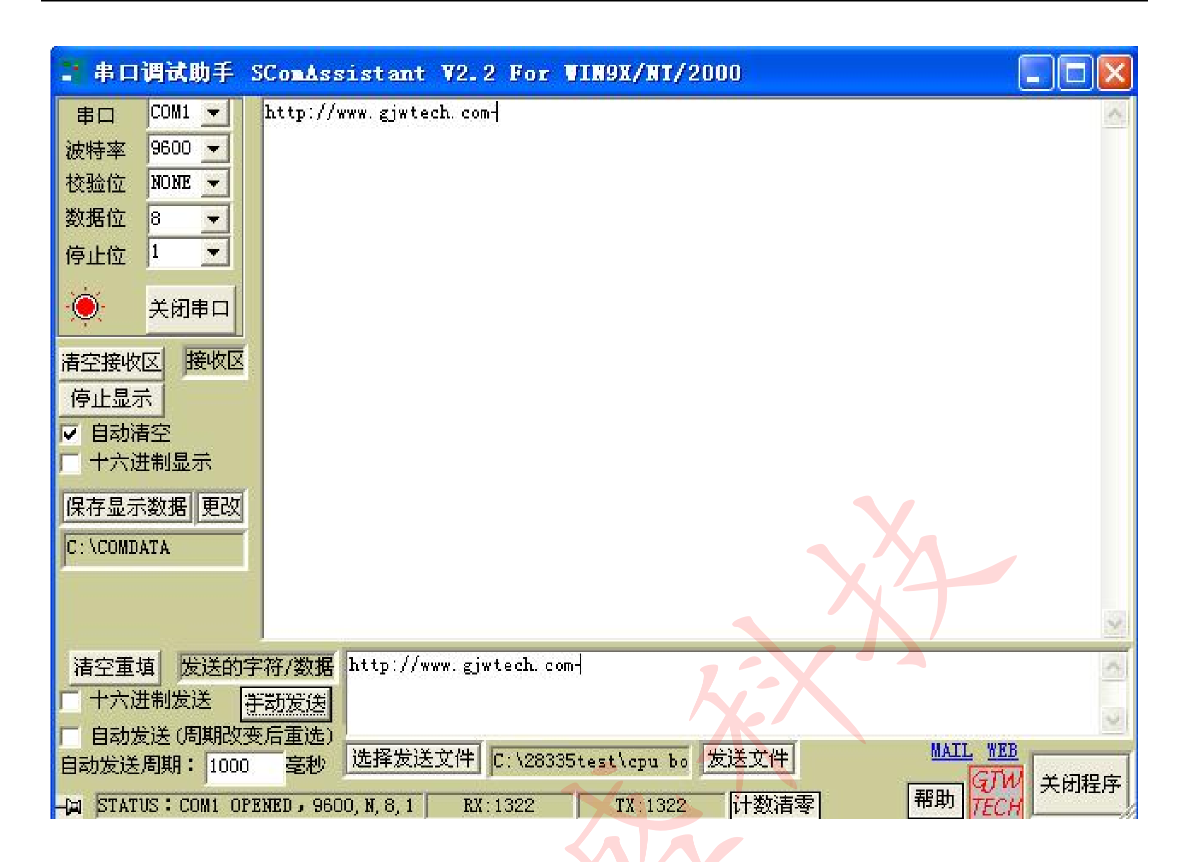
板“J10”串口单元的串行接口；并检查计算机、DSP 仿真器、实验箱是否正确连接。

2. 运行 CCS3.3 软件，调入样例程序，装载并运行；(进入 CCS 界面后需要“Debug--Connect”)
打开系统项目文件 \......\ CPU board\4 URAT Test\ example.pjt；
双击“example.pjt”及“Source”可查看各源程序；并加载“example.out”；

3. 运行“cpu board\examples\4 UART\uart 调试工具\串口调试软件”，出现界面后点击“设置”按钮，进行如下设置：端口根据计算机使用情况而选择



4. 设置完后，点击“手动发送，此时会在显示窗口中，显示如下：

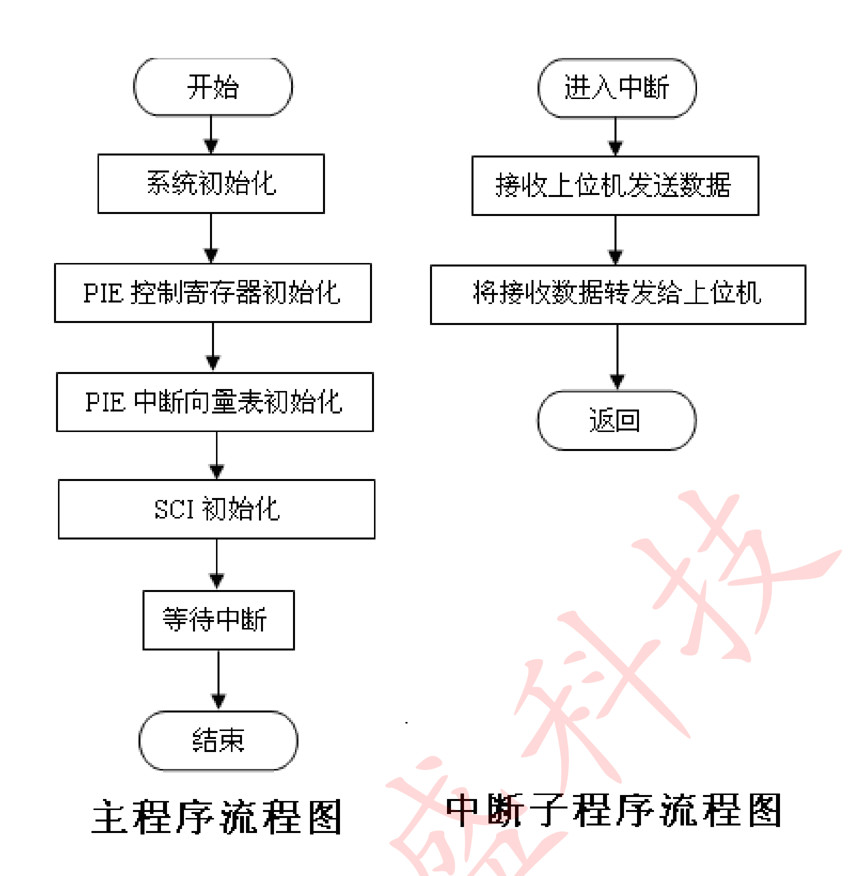


5. 关闭所有有关程序，该实验结束。

五、 实验说明

程序启用了 TMS320F28335 的串行口接收数据中断，当 DSP 通过串口接收到数据后在中断子程序中将接收到的数据原样通过 RS232 串口发送出。故此通过电脑端的串口测试程序发送一个数据，然后通过 DSP 传回，如果数据没变，则通讯成功。

六、 实验程序框图



主程序流程图　　中断子程序流程图

实验五 USB 实验

1：实验目的：

a 了解 DSP 开发系统的组成和结构

b 了解通过 usb 接口和 PC 机通讯的方法

2：实验器件

计算机（USB 端口为 USB2.0）、TechV28335 板、仿真器、usb A 口转 B 口线缆、CCS3.3 以上版本。

3：实验内容：

备注：该实验程序是在支持 USB2.0 端口的计算机条件下进行测试的。

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

4：实验步骤

1、运行 Code Composer Studio (CCS3.3)；

2、打开文件\.....\CPU board\5 USB Test\example.pjt；

3、编译全部文件并装载；

4、全速运行程序；

5、使用 usb 线浪连接电脑和 TechV28335 板子；

注意：如果是第一次使用该板子，电脑会提示没有找到驱动，需要安装驱动；进入文件夹..\CPU board\5 USB Test\Pc Driver 双击 SETUP.EXE 运行安装程序。弹出下图安装界面

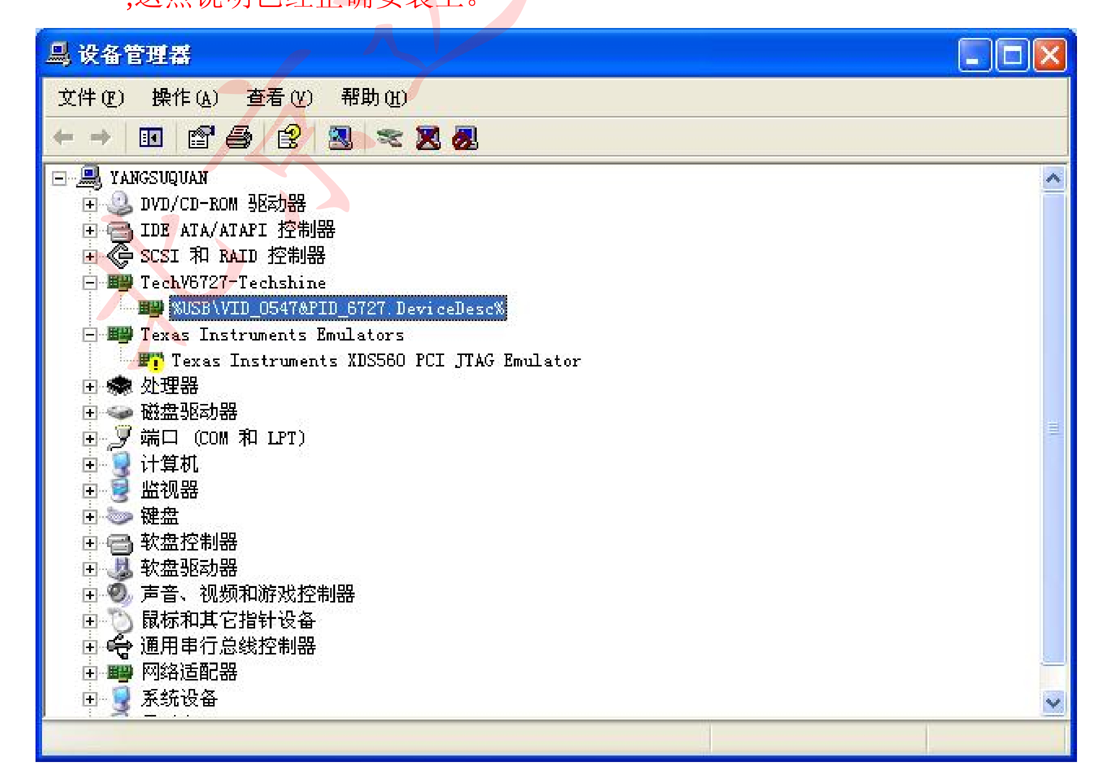


点击 INSTALL 等一会便会自动安装完成。

如果刚安装完成 USB 驱动，断开 USB 连线重先连接，打开电脑设备管理器可以看到设备已经找到：

说 明 ： 由 于 公 司 使 用 统 一 的 CY68013 测 试 驱 动 ， 故 在 下 面 中 显 示 “TechV6727-Techsine”

,这点说明已经正确安装上。

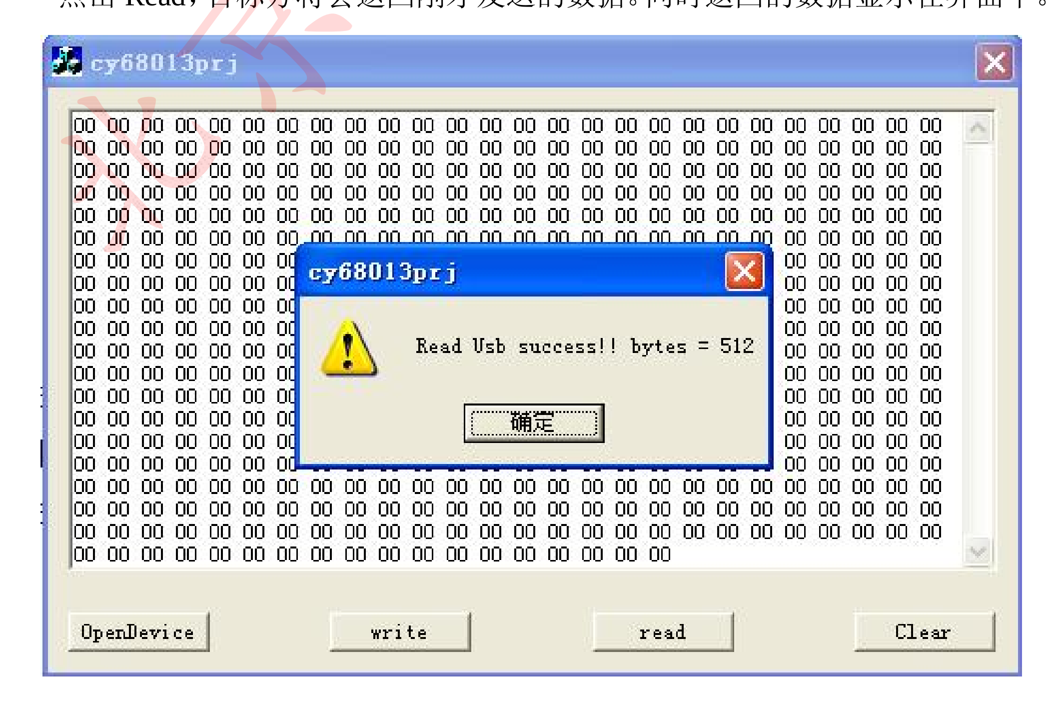


6、进入文件架..\CPU board\5 USB Test\PC app\Release 执行 cy68013prj.exe，弹出数据传输界面：



7、点击 OpenDevice 按钮，连接设备。设备连接好后，点击 Write 按钮（注意：必须程序在全速运行），PC 机将会向目标板发送 512 个 8bit 数据（00）。

点击 Read，目标办将会返回刚才发送的数据。同时返回的数据显示在界面中。



点击“确定”后再点击 Clear，清空显示区域内的数据，重复步骤（7），则在显示区内呈递增的趋势显示，直到上位机设定传输数据最高值后重新归 0 开始。

注意：PC 机和设备通讯时，需要先发送数据，才可以从设备中读取数据。



实验六 ECAN 实验

一、 实验目的

1. 学习 CAN 总线通讯原理；

2. C2000 系列 DSP 的 CAN 总线模块的操作方法；

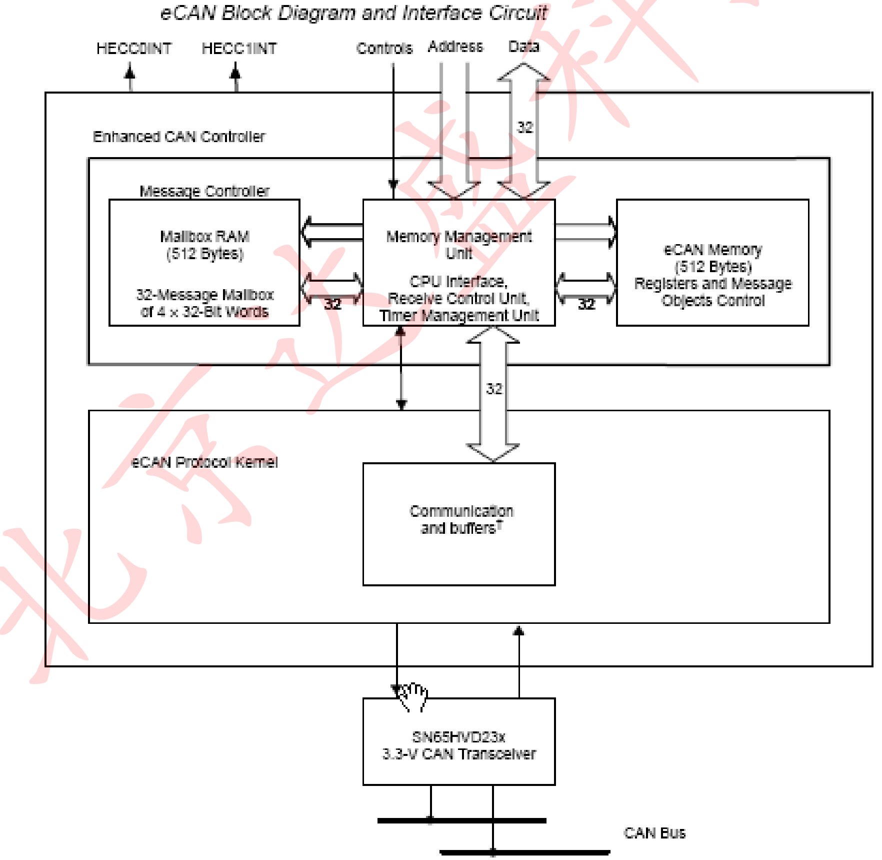
二、实验设备

计算机，CCS 3.3 版软件，DSP 仿真器，F28335CPU 板。

三、实验原理

在 F28335DSP 中使用的增强型控制器区域网络（eCAN）模块与现行的 CAN2.0 标准兼容。它可以使用已制定的协议在存在电子噪声的环境中与其他控制器进行串行通信。借助 32 个完全可配置的邮箱和时间标志特性，eCAN 模块提供了一种具有通用性和鲁棒性的串行通信接口。

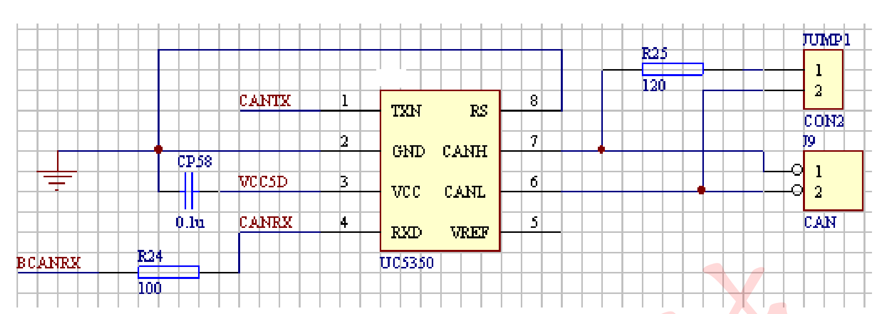
下面给出 eCAN 模块的功能框图：



eCAN 模块与用于 TI 公司的 24 系列微控制器的高端 CAN 控制器基本相同，而只作了一点修改。ECAN 模块与在 240x 系列 DSP 中的 CAN 模块相比，它增强了一些功能（如邮箱数目和时间标志等）。因此，为 240x CAN 模块所写的代码不能直接用于 eCAN。但是，eCAN 遵循了与 240x CAN 相同的

寄存器位设计结构和位功能，也就是说，许多寄存器和比特位在两种平台上具有完全相同的功能。这使得代码移植相对简单。

在本部分的设计中，CAN 总线接口经过 CAN 总线接线接口芯片 UC5350 与外部接口相连，原理图如下：



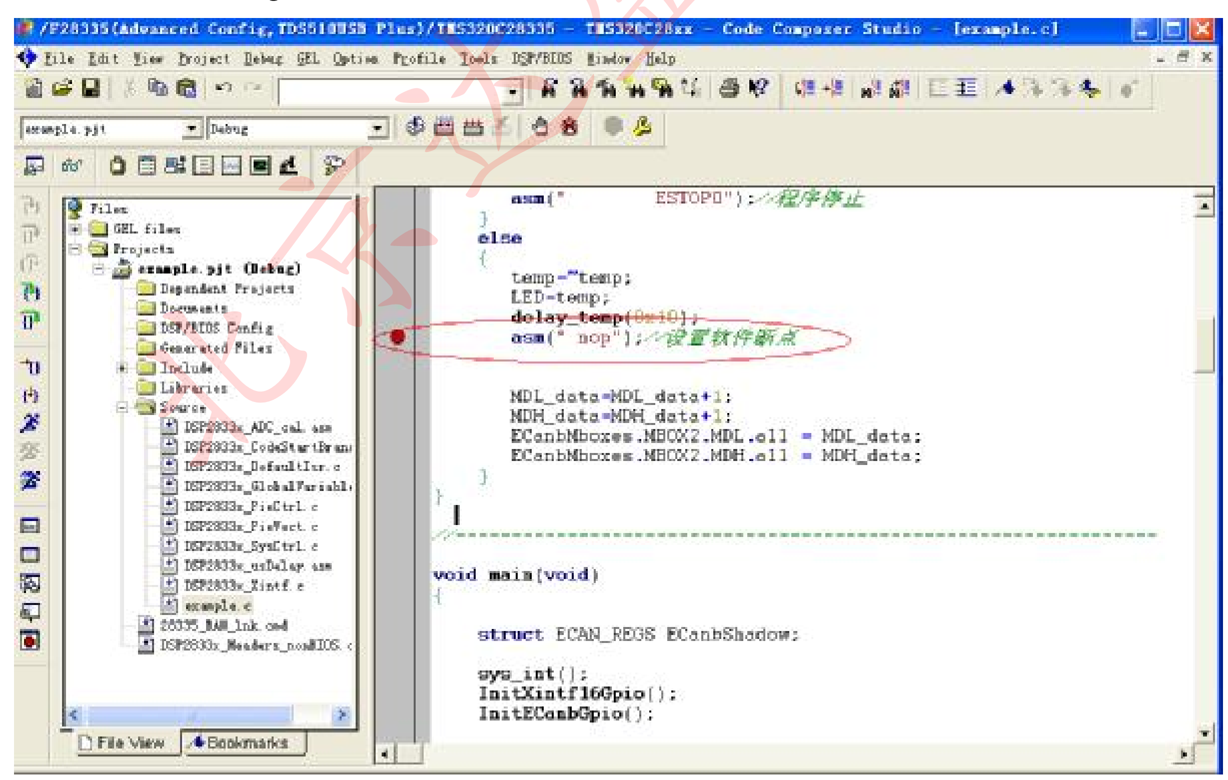
有关 eCAN 部分的寄存器请读者参阅相关的文档或书籍。

四、实验步骤

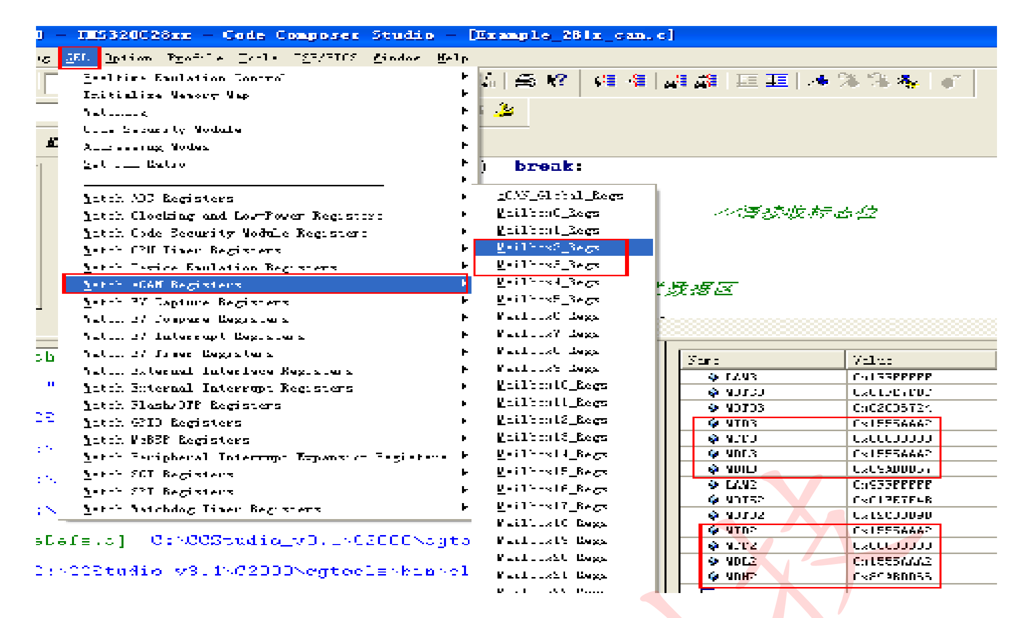
1. 运行 CCS 软件，调入样例程序，装载并运行；（进入 CCS 界面后需要“Debug--Connect”）
2. 用 Project/Open 打开“\cpu board\examples\6 ECAN\”工程文件；

   双击“example.pjt”及“Source”可查看各源程序；编译并加载 example.out”。
3. 点击“Debug \ Go main”进入主程序，在主程序中“m=0”处施加软件断点，

   单击“Debug \ RUN”,运行到断点处。



6. 单击“GEL \ Watch ECAN Registers”,在下拉菜单中，依次点击“Mailbox2_Regs”和“Mailbox3_Regs”，此时将两者加入观察窗口中，操作如下图：



7．点击“Debug\ Animate”全速运行程序,可以观察到观察窗口中：邮箱 3 和邮箱 2 的变化，两者之间的相对应邮箱内的数据一样，说明邮箱发送和接收成功。

Name	Value	Type	Radix
eCANB LAM2	0x098DD474	long	hex
eCANB MOTS2	0x00004575	long	hex
eCANB MOTO2	0x91DBEA95	long	hex
eCANB MID2	0x9555AAA2	long	hex
eCANB MCF2	0x910CCFA8	long	hex
eCANB MDL2	0x15000001	long	hex
eCANB MDH2	0x00000001	long	hex
eCANB LAM3	0x00000000	long	hex
eCANB MOTS3	0x00004575	long	hex
eCANB MOTO3	0x527CF415	long	hex
eCANB MID3	0x9555AAA2	long	hex
eCANB MCF3	0x00000008	long	hex
eCANB MDL3	0x15000001	long	hex
eCANB MDH3	0x00000001	long	hex

8．单击“Debug \ Halt”,停止程序运行。

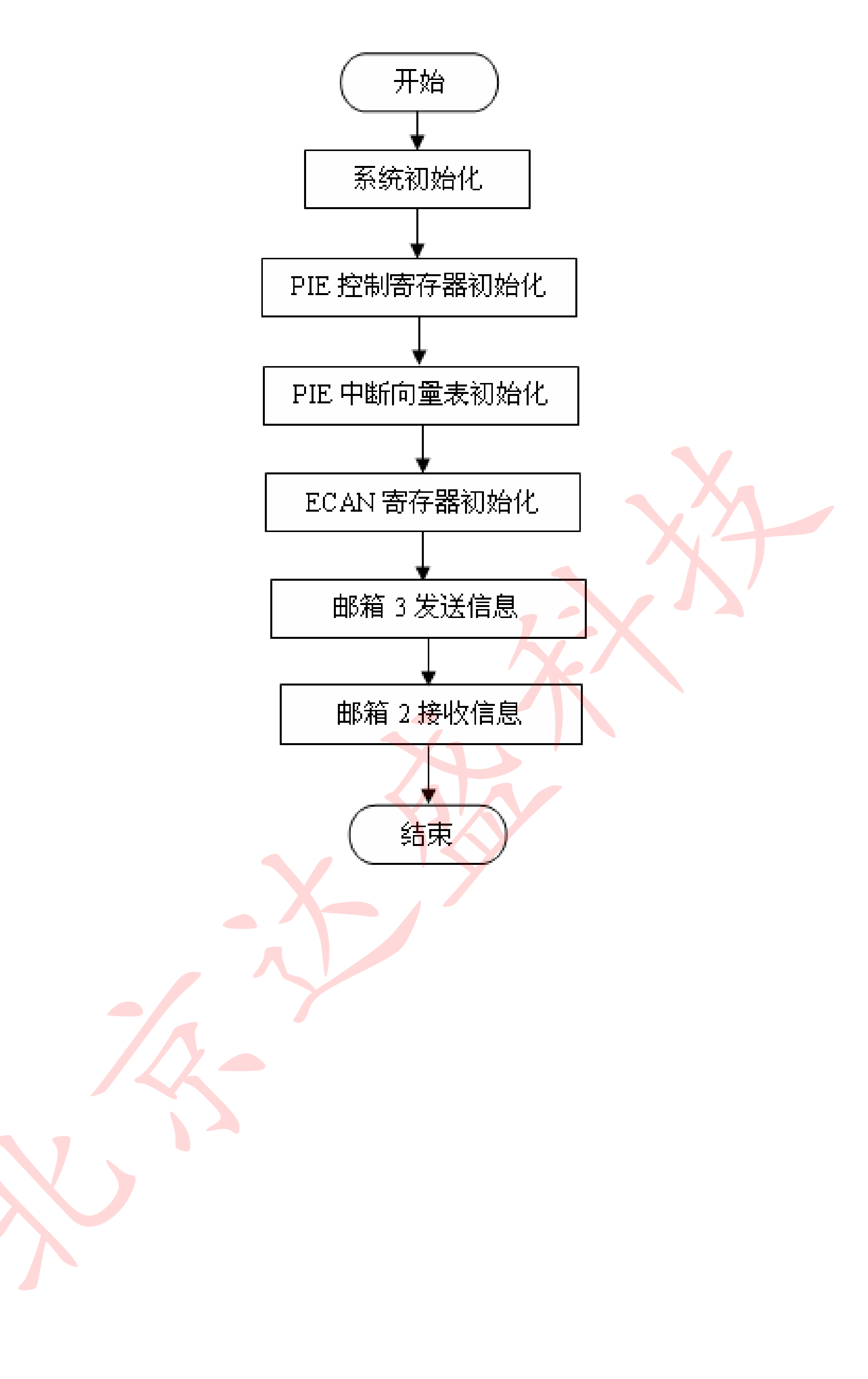
9．关闭所以实验窗口，本实验结束。

五、实验说明

本样例实验是使 CAN 控制器工作在自测试模式，在这种模式下，CAN 控制器能自己产生应答信号，信息帧没有真正发送出去，但是信息帧被读回，并储存在相应的邮箱中.

样例实验中的 CAN 控制器的邮箱 2 配置为接收方式，邮箱 3 配制为发送方式，发送和接收成功采用查询方式。（具体有关内容请参看 TI 的数据手册）

六、程序框图


开始
系统初始化
PIE 控制寄存器初始化
PIE 中断向量表初始化
ECAN 寄存器初始化
邮箱 3 发送信息
邮箱 2 接收信息
结束

实验七 PWM 波形产生实验

一、 实验目的

1. 了解 TMS320F28335 芯片的 EPWM 功能；
2. 掌握 AQ 模式产生 PWM 波的方法；

二、实验设备

计算机，CCS 3.3 版软件，DSP 仿真器，示波器。

三：实验内容：

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

四、实验说明

本样例实验是利用（Action-Qualifier）AQ 模式来产生两组（EPWM1A 和 EPWM1B）不同占空比的 PWM 波形，该模式是产生 PWM 波形的重要方式。

AQ 模式的功能框图如下：

Figure 2-17. Action-Qualifier Submodule

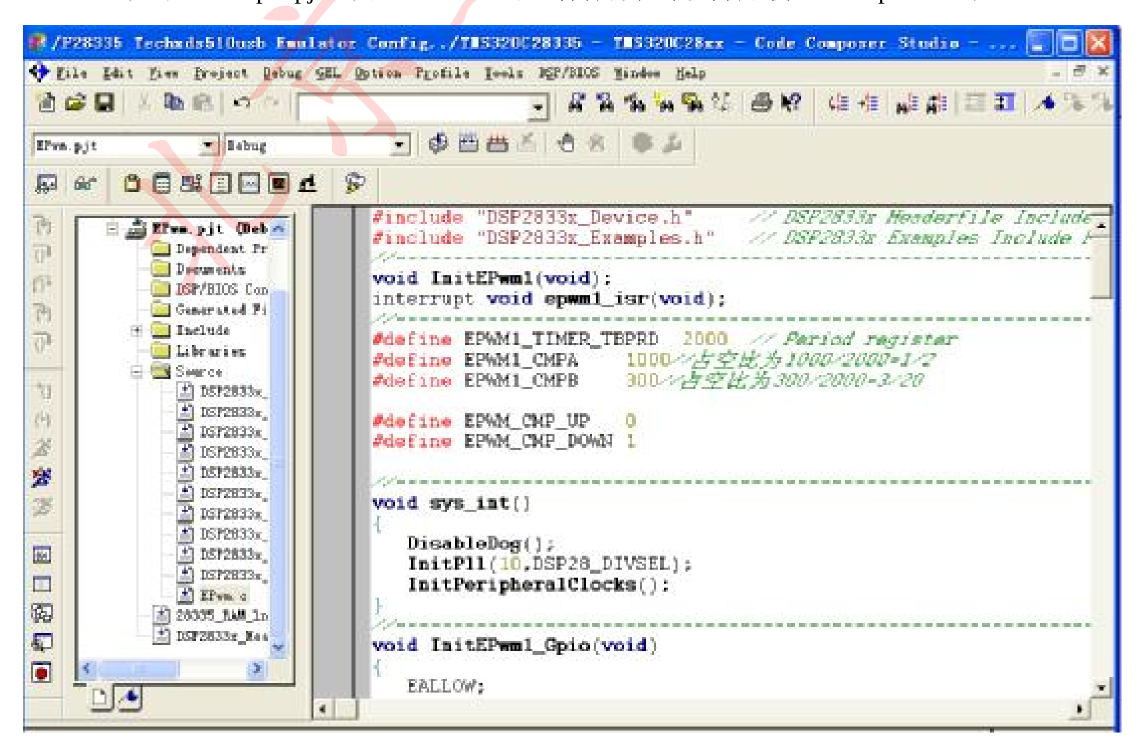
具体的原理应用请参看 TI DATESHEET 和实验程序。

五、实验步骤

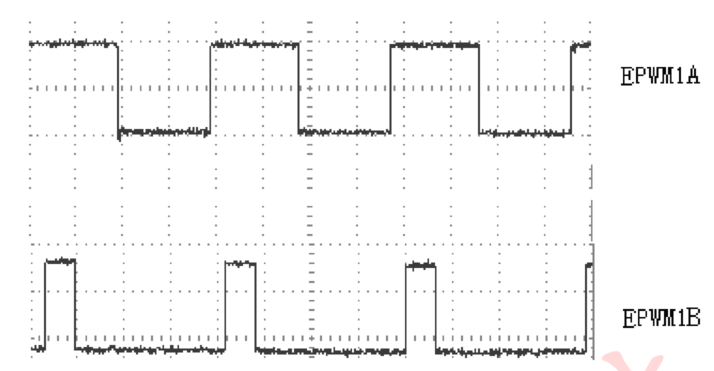
1、运行 CCS 软件，调入样例程序，装载并运行；（进入 CCS 界面后需要点“Debug--Connect”）

2 打开系统项目文件\\cpu board\examples\8 PWM\example.prj

4、 双击“example.pjt”及“Source”可查看各源程序；并加载“Example. out”；



5、 单击“Debug\ Run”运行程序，然后用示波器观察 F28335CPU 板上（U4）EPWM1A 和 EPWM1B 管脚上的输出波形(如下图)。



6、 关闭“example.pjt”工程文件；关闭所有窗口，本实验完毕。

六、程序修改说明

- 修改 PWM1A_duty_cycle 和 PWM1B_duty_cycle 的宏定义值，则可以精确的改变 EPWM1A 和 EPWM1B 的占空比。
- 本实验仅产生了 EPWM1A 和 EPWM1B 两路 PWM 波形，如果想对 EPWM2A，EPWM2B；EPWM3A，EPWM3B—EPWM6A，EPWM6B 产生各种占空比的 PWM 波形，请参看本样例程序进行修改设计

实验八 FLASH 烧写实验

一. 实验目的

3. 掌握将程序烧进 28335 的 FLASH 的方法;

二、实验设备

计算机，CCS 3.3 版软件，DSP 仿真器，示波器。

三：实验内容：

a 连接仿真器与用户开发板

在进行 DSP 实验之前，先必须连接好仿真器与系统控制板，连接的方法如下所示：



b 运行 CCS 程序

●成功运行程序后，首先应熟悉 CCS 的用户界面

●学会 CCS 环境下的程序编写、调试、编译、装载，学习如何使用观察窗口等。

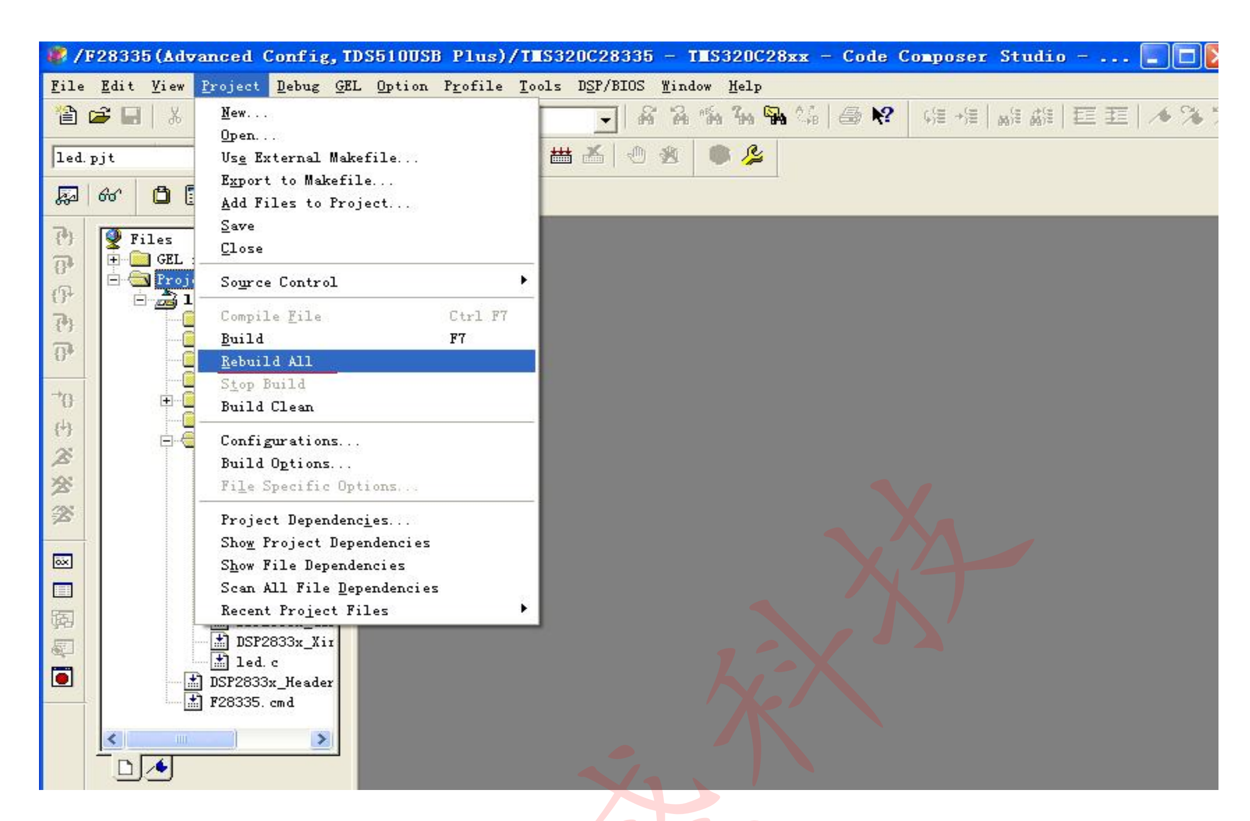
四、实验步骤

1、运行 CCS 软件，调入样例程序，装载并运行；（进入 CCS 界面后需要点“Debug--Connect”）

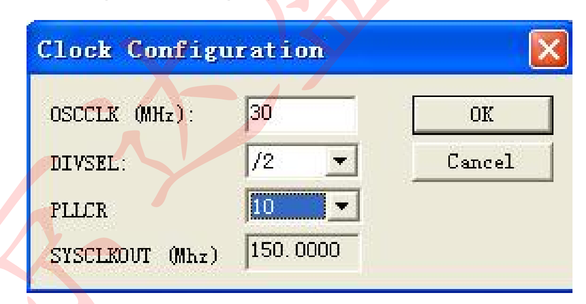
2 打开系统项目文件\\cpu board\ flash burn\examples\led.prj

7、 双击“led.pjt”及“Source”可查看各源程序

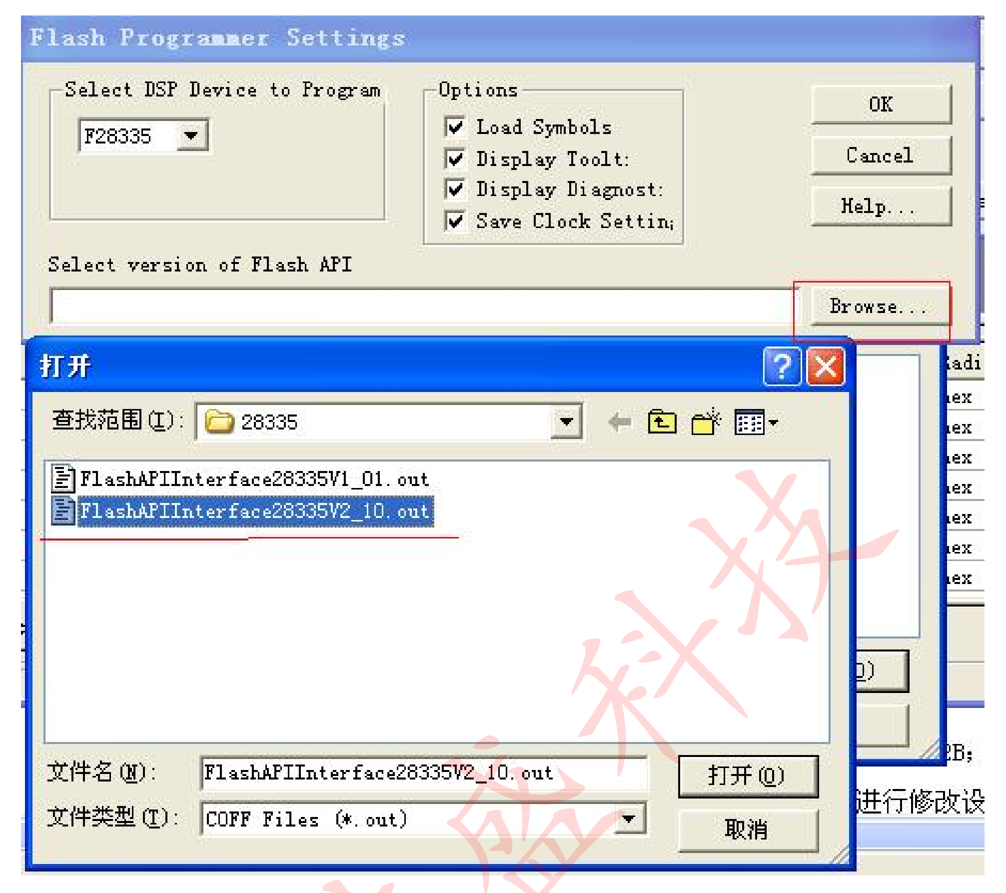
8、 单击“PROJECT—REBUILD ALL”，全部编译文件生成可烧写的“.out”文件



9、 点击 TOOL/F28XX on-chip FLASH programmer，设置如下：

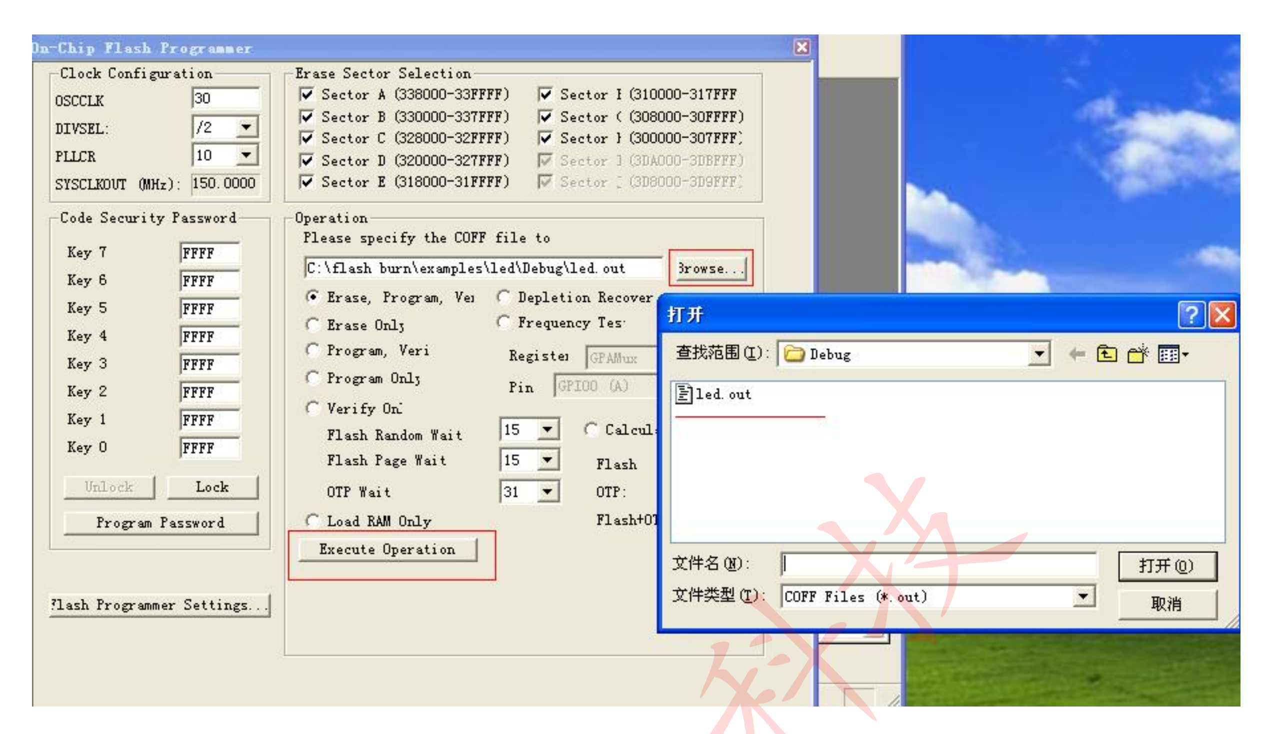


点击“OK”，然后在下面界面中点击“BROWSE”，选择画横线的部分，



点击“OK”，在下面的界面中进行设置：

点击“BROWSE”，选择好要要烧写的的“.OUT”文件，然后点击红色标框内钮，此时烧写插件会将要烧写的程序，烧进内部 FLASH 中。



烧写成功后，断电，将开发板和仿真器脱离，然后进行 SW2 开关的设置：

开关 SW2（1：OFF；2：OFF；3：OFF；4：OFF），

然后：

单独给 CPU 供电，按动一下 CPU 板子上复位键，此时会观察到开发板的 LED5-LED8 灯，依次闪烁变化

10。程序到此烧写成功。